

Martti Hyssälä

**TAPAUSTUTKIMUS
OHJELMISTOPROJEKTIN
LÄPINÄKYVYYDESTÄ
ASIAKASPORTAALIN
KEHITYSPROJEKTISSA**

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Huhtikuu 2020

TIIVISTELMÄ

Martti Hyssälä: Tapaustutkimus ohjelmistoprojektin läpinäkyvyydestä asiakasportaalin kehitysprojehtissa

Tampereen yliopisto

Diplomityö

Huhtikuu 2020

Johtamisen ja tietotekniikan DI-tutkinto-ohjelma

Pääaine: Ohjelmistotuotanto

Tarkastaja: Apulaisprofessori Sami Hyrynsalmi, Professori Marko Seppänen

Ketterien menetelmien ohjelmistoprojektit yleistyvät nopeasti. Ne ovat tehokkaita huomioimaan asiakkaan mielipiteitä useasti tekemisen aikana. Eräs yleisimmistä ketteristä menetelmistä on Scrum. Siinä tekeminen pilkotaan pieniin iteraatioihin, sprintteihin. Jokaisen iteraation päätyttyä valmiit kehitystarinat esitellään asiakkaille ja asianomistajille iteraation katselmointikokouksessa. Asiakkailta saadaan näin heti palautetta valmiista toteutuksesta. Palautetta voidaan heti hyödyntää seuraavien iteraatioiden suunnittelussa.

Tässä tapaustutkimuksessa selvitettiin asiakasportaaliprojektin viestinnän onnistumista globaalissa yrityksessä. Tapausprojekti oli ensimmäisiä ison mittaluokan projekteja kohdeyrityksessä, jossa hyödynnettiin Scrum-viitekehystä projektin toteuttamiseksi. Projektilla oli iso joukko asianomistajia organisaation eri tasoilla. Tämän tutkimustyön tarkoitus oli tutkia, saavatko projektia seuraavat tahot riittävästi tietoa projektin tavoitteista, tähänastisista saavutuksista sekä projektin aikaansaannoksista.

Tämä tapaustutkimus suoritettiin valikoitujen haastattelujen avulla siinä vaiheessa, kun projektin aloittamisesta oli kulunut ajallisesti noin yksi vuosi. Tuona hetkenä projektissa oli suoritettu yhteensä 26 iteraatiota. Tutkimukseen haastateltiin yhteensä seitsemän henkilöä. Haastatteluihin valittiin sopiviksi katsottuja henkilöitä eri puolelta organisaatiota. Haastattelut pidettiin puolistrukturoidusti, kysymyslistaa hyväksikäyttäen. Haastattelujen avulla tutkimukseen kerättiin kvalitatiivinen data. Tutkimuksen kvantitatiivinen data kerättiin iteraation 27. julkaisun yhteydessä. Tässä yhteydessä tutkimuksessa saatiin selville organisaation aktiivisuus eri viestintäkanavissa.

Haastattelututkimuksessa selvisi, että tapausprojektin läpinäkyvyyteen oltiin tyytymättömiä, osittain tyytyväisiä ja tyytyväisiä. Osa haastateltavista oli halukkaita käymään pitkiä keskusteluja tyytymättömyyden syistä ja odotusarvoista. Etenkin näistä, kritiikkiä sisältäviä haastatteluja pyrittiin analysoimaan, pohtimaan ja tunnistamaan kehitysehdotuksia. Viestintäkanavista Yammer oli käyttäjämäärällisesti suosituin. Osa haastateltavista kuitenkin piti myös katselmointikokouksia tärkeinä mahdollisuutena vaikuttaa projektiin. Tämän tutkimuksen tuloksia voidaan hyödyntää uusien projektien kommunikointia suunniteltaessa.

Avainsanat: Scrum, ketterä kehittäminen, ohjelmistokehitys, projektiviestintä.

ABSTRACT

Martti Hyssälä: Case study on Scrum project's transparency in a global organization

Tampere University

Master of Science Thesis

April 2020

Master's Degree Programme in Management and Information Technology

Major: Software Engineering

Examiners: Associate Professor Sami Hyrynsalmi, Professor Marko Seppänen

Software projects lead by agile principles have become de facto development way rather quickly in industry. The agile methods are suitable for listening customer needs constantly during the project journey. Nowadays, Scrum is one of the most used in agile methods. In Scrum project, the scope is split in small pieces, called stories, and development is done in sprints. After every sprint, results are shown to the project stake holders in demonstration meeting. The project's stakeholders can then give instant feedback on results and Scrum team can use feedback when planning upcoming sprints.

This thesis was a case study on transparency in a Scrum project in a global company. The case project was one of the first big-scale projects utilizing scrum in the case company. The project had several internal stakeholders in different levels of organization. The objective of this study was to address the question whether the project stakeholders are getting enough information on project outcomes so far.

This case study was done by interviews. Interviews were done in a point when project had been running one year from the beginning (26 iterations). The seven interviewees were selected by their place in the organization and by their position in the case project. The interviews were done in semi-structured way, utilizing a list of questions to give the interview a structure but not limiting discussion. From these interviews the qualitative data was collected for this study. Quantitative data was collected when 27th iteration was completed. Quantitative data is used to find out the volume of interested participants in different communication channels.

In this case study, it was found out that some of the interviewed were not satisfied with the project's transparency, some were partially satisfied and some totally satisfied. Some of the interviewees were eager to give long and fruitful feedback on what could have been done differently. It was also found out that Yammer is the most popular tool to follow projects in the case organization. However, there is also great demand for interactive demonstration meetings. These learnings can be used in the case company to improve the transparency in the case project as well as in upcoming Scrum projects.

Keywords: Scrum, agile project, software development, software project communication.

ALKUSANAT

Tätä diplomityötä ei olisi syntynyt ilman tapaustutkimuksen kohdeyritystä, ja sen oppimishaluista organisaatiota. Haluan kiittää siis yritystä mahdollisuudesta, tuesta ja avusta. Kaikki haastattelut sujuivat avoimessa ilmapiirissä, kiitos haastatelluille tutkimusdatan tuottamisesta. Iso kiitos, Paavo, ohjaamisesta, neuvoista, avuista, tarvittaessa patistamisesta ja ylipäättään koko työn mahdollistamisesta! Kiitos myös ohjaajalleni, apulaisprofessori Sami Hyrynsalmelle avusta ja ohjaamisesta, osasit antaa aina selkeät ohjeet työn eteenpäin viemiseksi juuri silloin, kun niitä tarvitsin.

Turussa, 14.4.2020

Martti Hyssälä

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. OHJELMISTOTUOTANTO.....	3
2.1 Ohjelmistotuotannon perinteiset menetelmät	3
2.2 Agile Manifesto	5
2.3 Ohjelmistotuotannon haasteet.....	7
2.4 Tutkimuksia ohjelmistotuotannon haasteista	7
3. SCRUM.....	12
3.1 Scrum-prosessi	13
3.2 Scrum-roolit	15
3.3 Scrum-prosessin tapahtumat	16
3.4 Scrum-kommunikaatio	18
3.5 Scrum isossa organisaatiossa.....	18
4. LÄHTÖKOHTA JA TUTKIMUSPROSESSI	23
4.1 Tutkimuksen lähtökohta	23
4.2 Tutkimuksen kohdeyritys ja sen organisaatio	23
4.3 Menetelmät	26
4.4 Tiedon analysointi	28
4.5 Tutkimuksen luotettavuus ja yleistettävyyys	29
5. KYSELYTUTKIMUKSEN TULOKSET	30
6. ANALYYSI JA JOHTOPÄÄTÖKSET	36
7. YHTEENVETO.....	41
LÄHTEET	44
LIITE A: 12 ALKUPERÄISTÄ TEESIÄ AGILE MANIFESTO ARVOJEN TAKANA ..	46
LIITE B: HAASTATTELUN MUISTILISTA SUOMEKSI.....	47
LIITE C: HAASTATTELUN MUISTILISTA ENGLANNIKSI	48

KUVALUETTELO

- Kuva 1.** Ohjelmistokehityksen vesiputousmalli. Sivu 4.
- Kuva 2.** Vesiputousmalli inkrementaalisuutta hyödyntäen. Sivu 5.
- Kuva 3.** Tilasto ohjelmistoprojektien onnistumisesta vuosina 2011 – 2015. Sivu 8.
- Kuva 4.** Ketterä versus vesiputousmalli jaettuna projektien koon mukaan 2011-2015. Sivu 9.
- Kuva 5.** Ketterien tiimien kyvykkyys ja projektien onnistuminen 2011-2015. Sivu 10.
- Kuva 6.** Scrum-prosessi. Sivu 13.
- Kuva 7.** Scrum-iteraatio. Sivu 16.
- Kuva 8.** Kettusen ja Laantin ketterän yrityksen konsepti. Sivu 19.
- Kuva 9.** Leffingwellin ketterän yrityksen konsepti. Sivu 21.
- Kuva 10.** Tapausprojektin rakenne. Sivu 25.

1. JOHDANTO

Ohjelmistojen hyödyntäminen on nykyisin tärkeä liiketoiminnan osa lähes jokaiselle yritykselle. Ohjelmistoilla hoidetaan esimerkiksi raportointia, tuotannonohjausta, asiakaspalvelua tai markkinointia. Useaan eri tarpeeseen löytyy valmiita ohjelmistoja ja tietojärjestelmiä. Valmiit ohjelmistopakettit ovat usein konfiguroitavissa tarpeeseen sopivaksi. Toisinaan yrityksen tarve ohjelmistolle on hyvin yksityiskohtainen, tai käyttötarve on jotakin uutta markkinoilla. Toisinaan ohjelmisto pitää rakentaa itse täyttämään tätä tiettyä tarvetta. Tällöin puhutaan ohjelmiston kehityshankkeesta. Ohjelmistokehityshankkeen hallittuun läpiviemiseen on olemassa erilaisia projektimalleja. Lineaarinen vesiputousmalli edustaa perinteistä ohjelmistoprojektimallia, ja on mahdollisesti edelleen eräs yleisimmin käytetyistä ohjelmistoprojektimalleista. Viimeisen kahden vuosikymmenen aikana vesiputousmallin rinnalle on yleistynyt myös erilaisia ketterän kehittämisen malleja.

Ketterät työskentelytavat ohjelmistoprojekteissa ovat yleistyneet nopeasti viimeisten vuosien aikana. Suuressa globaalissa yrityksessä ketterän toimintatavan käyttöönotto tapahtuu usein ensin organisaation sisällä yhdessä tai useammassa tiimissä. Tästä on kuitenkin pitkä matka koko organisaation kehittymiseen ymmärtämään prosesseja ja toimimaan ketterästi. Laaja globaali organisaatio, useat eri liiketoiminta-alueiden sisäiset edustajat sekä useat ulkoiset asiakkaat, ovat haaste projektikommunikoinnille.

Ketterien projektimallien, joista esimerkkinä on Scrum, tärkeitä tavoitteita ovat läpinäkyvyys projektin asiakkaille ja asianomaisille, nopea reagointikyky muutoksiin projektin sisällössä, keskittyminen tuottamaan lisäarvoa asiakkaalle mahdollisimman nopeasti ja projektin mahdollisimman kevyt hallinto. Läpinäkyvyys (transparency) on tärkeää niin Scrumissa, kuin muissakin ketterissä projektimenetelmissä. Läpinäkyvyyttä on kuitenkin tutkittu nykyisessä kirjallisuudessa niukasti.

Vastatakseen tähän tutkimusalueessa olevaan puutteeseen, tämä diplomityö esittää tapaustutkimuksen Scrum-menetelmällä tapahtuvan ohjelmistoprojektin läpinäkyvyyden onnistumisesta. Erityisesti tässä työssä etsitään vastausta kysymykseen *miten Scrum-projektin asianomaiset kokevat läpinäkyvyyden toteutuneen?* Tässä yhteydessä

läpinäkyvyydellä tarkoitetaan sitä, miten hyvin projekti on onnistunut kommunikoimaan ulkopuolisille tahoille projektin etenemisestä.

Kohdeyritys on kansainvälinen metalli-, ja teknologiateollisuudessa toimiva, koneita ja laitteita valmistava, ja myymiensä tuotteiden ja palveluiden koko elinkaarta tukeva yritys. Tutkimuskohteen projektissa sovellettiin Scrum-menetelmää kyseisessä organisaatiossa. Läpinäkyvyyden nykytasoa selvitetään puolistrukturoiduilla haastatteluilla organisaation eri tasoilla projektiin osallistuvien tahojen kanssa.

Loput tästä työstä on jäsennelty seuraavalla tavalla. Luvussa 2 esitellään Scrumia, Scrumin ja ketterien menetelmien implementointia suuriin organisaatioihin, projektikommunikointia ja organisaation kehittymistä ketteräksi. Luvussa 3 kerrotaan kohdeorganisaatiosta, sen käyttämistä työkaluista ja tapausprojektin toimintatavasta. Luvussa 4 käydään läpi tutkimuksen lähtökohta, tapaus tutkimuksen organisaatio, tutkimusmenetelmät, tiedon analysointi ja tutkimuksen luotettavuus. Luvussa 5 esitellään kyselytutkimuksen tulokset. Kappale 6 sisältää kyselytutkimukseen perustuvan analyysin ja johtopäätökset. Kappaleesta 7 löytyy tutkimuksen yhteenveto.

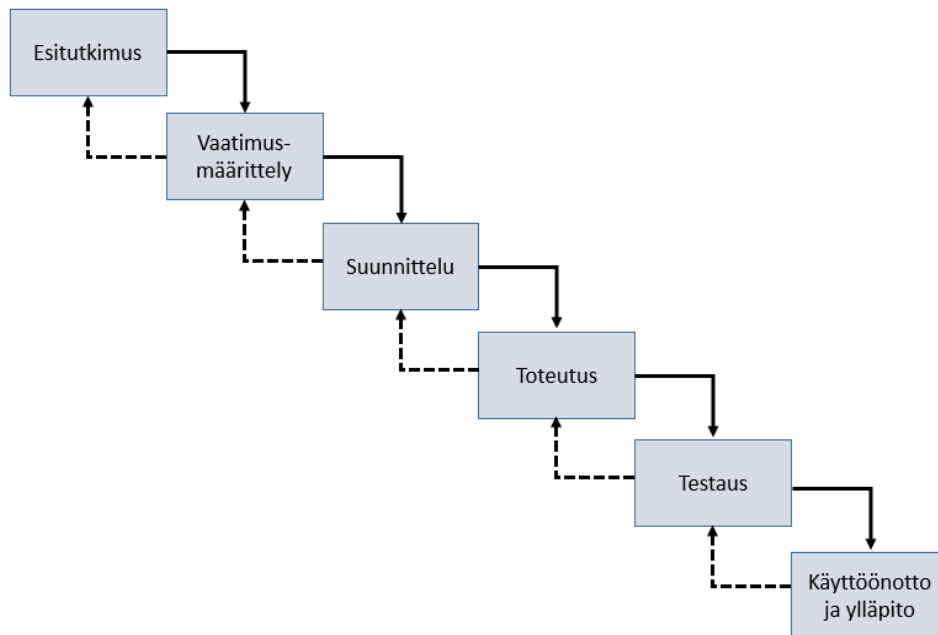
2. OHJELMISTOTUOTANTO

Ketterä kehitys on yksi ohjelmistokehityksen muoto. Ketterässä kehittämisessä hyödynnetään inkrementaalisuutta, kokonaisuus toimitetaan vähän kerrallaan pienissä erissä. Ketterässä kehittämisessä hyödynnetään myös iteratiivisuutta, sovellus kehitetään kehityssykleissä. Merkkejä iteratiivisten ja inkrementaalisten metodien käyttämisestä yleisellä tasolla projektinkehityksessä löytyy jo vuodelta 1957. Sopeutuva ohjelmistonkehitys ja iteratiivinen projektin hallinta yleistyivät 1970-luvun alkupuolella. Sen sijaan, että projektisisällön suunnitteluun käytetään paljon aikaa ja resursseja ennen kehitystyön alkua, ketterässä projektissa suunnittelua tehdään projektin aikana, yrittäen saada pieniä konkreettisia tuloksia mahdollisimman nopeasti. Isoin muutos ohjelmistoprojektikehityksen kannalta tapahtui kuitenkin vasta 90-luvulla. Usean uuden ja kevyen projektinhallintamenetelmän ohella alkunsa sai myös nykyisin suosittu Scrum [1].

Tässä luvussa perehdytään tutkimuskysymyksen taustoihin. Jotta voimme ymmärtää ketterän ohjelmistoprojektin läpinäkyvyyttä meidän pitää ymmärtää minkälainen on ketterä ohjelmistoprojekti ja mitä tarkoittaa Scrum.

2.1 Ohjelmistotuotannon perinteiset menetelmät

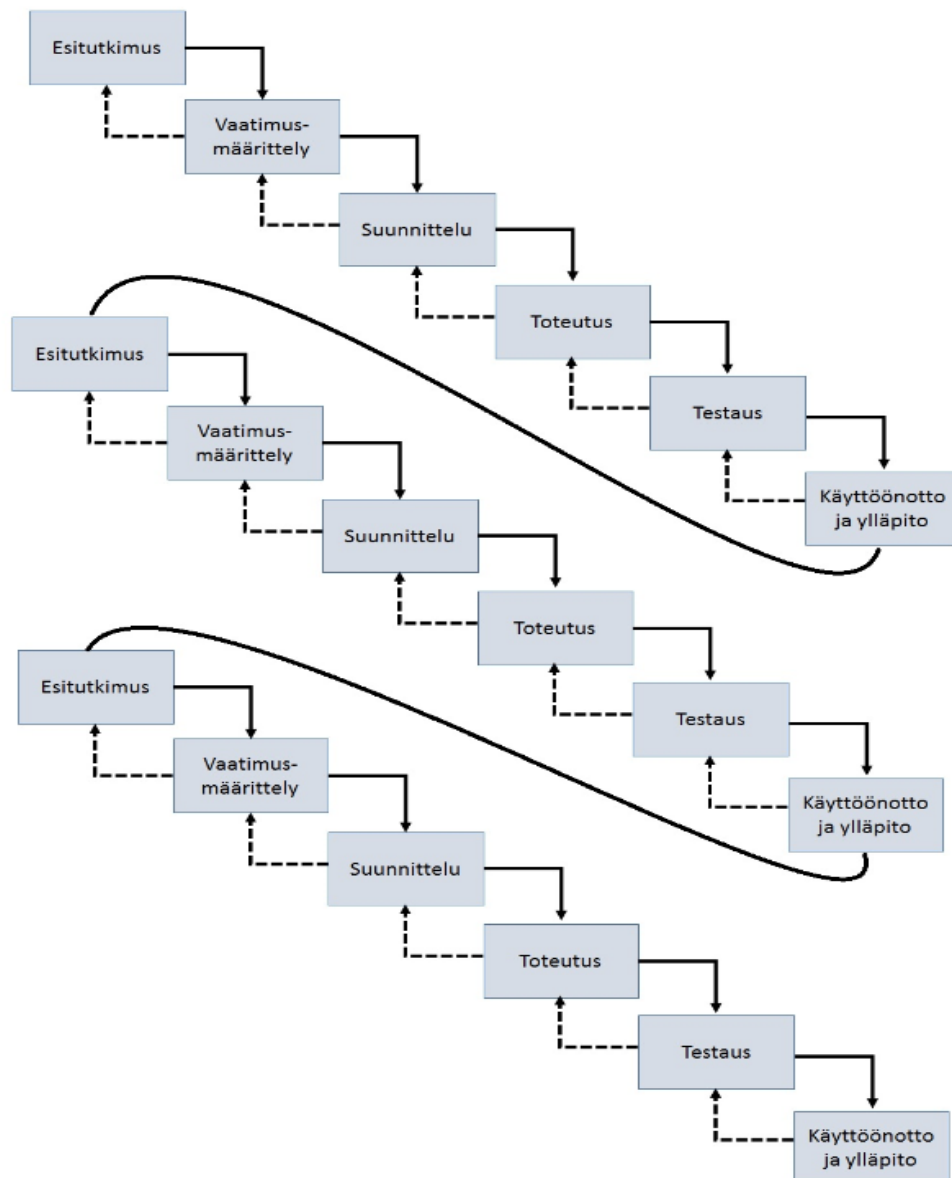
Ohjelmistoprojektien hallittuun läpivientiin on olemassa useampia projektimalleja. Eräs vanhimmista ja edelleen yleisesti käytössä oleva on vesiputousmalli. Sen kehitti ja esitteli Winston Royce artikkelissaan vuonna 1970. Vesiputousmalli on yksinkertainen ja lineaarinen. Vesiputousmallissa edetään portaittain vaiheesta toiseen, projektin alusta sen loppuun asti. Jokaisen vaiheen lopussa tarkastellaan, ollaanko valmiita jatkamaan seuraavaan vaiheeseen vai tulisiko vaihetta jatkaa ennen siirtymistä seuraavaan. Katselmointi ennen seuraavaa vaihetta on tärkeää, sillä lineaarisessa projektimallissa ei ole mahdollisuutta palata edellisiin vaiheisiin [2]. Vesiputousmallissa oletetaan, että vaatimukset voidaan määritellä etukäteen, eikä niihin tule enää projektin aloituksen jälkeen merkittäviä muutoksia. Käytännössä tämä tarkoittaa sitä, että käyttäjät tietävät etukäteen, mitä he haluavat, ennen kuin näkevät lopputuloksen. Muun muassa tästä johtuen noin 29% vesiputousmallin projekteista on epäonnistunut [3].



Kuva 1. Ohjelmistokehityksen vesiputousmalli (muokattu lähteestä [2]).

Alkuperäisestä Roycen vesiputousmallista on kehittynyt vuosien varrella hieman erilaisia variaatioita. Kuvassa 1 on eräs versio 6-portaisesta vesiputousmallista. Tässä vesiputousmalli antaa mahdollisuuden palata edelliseen vaiheeseen tekemään muutoksia muuttuneiden tarpeiden tai opittujen asioiden perusteella. Vesiputousmallissa ei kuitenkaan voida siirtyä uuteen vaiheeseen ennen kuin edellinen vaihe on saatu valmiiksi. Mikäli jossakin vaiheessa ilmenee tarpeita muutokselle alkuperäiseen suunnitelmaan, tulee prosessissa edetä taaksepäin vaihe vaiheelta. Vaiheen ohittamista ei sallita. Asiakastarve tulee täytetyksi vasta aivan prosessin lopussa, käyttöönotossa. Vesiputousmalli sopii parhaiten ohjelmistoprojekteihin, joissa on erittäin selkeät vaatimukset, eikä riskiä vaatimusten muutoksille myöhäisemmissä projektin vaiheissa ole [4].

Vesiputousmallia voidaan käyttää myös inkrementaalisesti (kuva 2). Tällöin isompi kokonaisuus jaetaan pienempiin osiin ja toteutetaan pienissä paloissa. Inkrementaalisessa kehitysmallissa voidaan toteuttaa useampia vesiputousmallin projekteja peräkkäin tai rinnakkain. Inkrementaalinen vesiputousmalli on askel ketterän kehittämisen suuntaan, tulevaisuudessa on mahdollista reagoida ilmenneisiin muutostarpeisiin [5].



Kuva 2. Vesiputousmalli inkrementaalisuutta hyödyntäen (muokattu lähteestä [5]).

2.2 Agile Manifesto

Ketterän projektinhallinnan juuret ulottuvat 1990-luvun keskivaiheille. Joukko kokeneita ohjelmiston kehittäjiä työskenteli rinnakkain usean eri projektin, ohjelmointikielen ja vaatimussisällön parissa. Työskentelyn yhteydessä he kokeilivat erilaisia työskentelymalleja ja toimintatapoja. Ajatuksena oli löytää ratkaisuja perinteisiin ohjelmistonkehitysongelmiin. Tavoitteena oli tuottaa laadukasta ohjelmistoa nopeasti, tehdä mahdollisimman vähän hukkaan menevää työtä ja välttää tarpeetonta kehittämistä liian aikaisessa vaiheessa [6].

Vuonna 2001 useat ketterien menetelmien kehittäjistä kokoontuivat yhteen keskustelemaan kokemuksistaan, ja pohtimaan millaisia yhteisiä toimintatapoja heillä

on. Lopputuloksena syntyi Agile Manifesto. Siinä yhdistyvät ketterien mallien samankaltaisuudet, arvomaailma ja tavoitteet [6].

Agile Manifeston yhteiset arvot ovat [1]:

- Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
- Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota
- Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja
- Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Painavammat arvot ovat arvolauseiden vasemmalla puolella. Arvoja tukemaan määriteltiin 12 teesiä [Liite 1].

Tärkeintä ketterässä kehittämisessä on tuottaa asiakkaalle käyttökelpoista, hyödyllistä ja käytettävyydeltään korkeatasoista ohjelmistoa, jatkuvilla julkaisuilla tasaisin väliajoin, ja mahdollisimman varhaisessa vaiheessa projektia. Tämän lisäksi arvioidaan tuottamisprosessin tehokkuutta itsearvioinnilla tasaisin väliajoin, ja kehitetään projektin prosesseja jatkuvasti paremmiksi [6]. Agile Manifeston tavoite on saavuttaa ketterän kehittämisen ajattelu- ja arvomaailma. Se ei kuitenkaan itsessään tarjoa varsinaisia työkaluja, prosesseja tai toimintatapoja ketterään kehittämiseen [7].

Katayama ja Bennett määrittelevät ketteryydelle neljä tukipilaria [8]:

- Sen pitää tuottaa arvoa loppuasiakkaalle
- Olla valmis ympäristön tai vaatimusten muutoksiin
- Arvostaa ja hyödyntää ihmisten tietotaitoa
- Luoda virtuaalisia kumppanuuksia

Gould määrittää ketteryyden olevan kyky toimia nopeasti tapahtuvissa ja vaikeasti ennustettavissa muutoksissa [9]. Koska määritelmät ketteryydestä vaihtelevat, ketteryyden ja Agile Manifeston konsepti on moninainen ja monimutkainen [7]. Seuraavassa luvussa käsitellään Scrum-viitekehystä, joka on eräs ketterän kehittämisen toimintamalleista.

2.3 Ohjelmistotuotannon haasteet

Ian Sommervillen mukaan perinteiset ohjelmistokehityksen haasteet ovat seurausta suurista, kriittisistä ja pitkäikäisistä ohjelmistojärjestelmistä, joita on ollut kehittämässä useita yksittäisiä kehittäjiä monissa eri kehitystiimeissä [10].

Perinteisesti toteutettu ohjelmistoprojekti koostuu yleensä yksityiskohtaisesta suunnitelmasta, muodollisesta laadunvarmistuksesta, useista eri työkaluista ja erittäin tarkasta kehitysprosessista. Suunnitellussa ja dokumentoinnissa syntyy näin päällekkäisyyttä, turhaa työtä, jota ei voi hyödyntää. Sommervillen [10] mukaan päällekkäisyys on kuitenkin perusteltua ison kokoluokan projekteissa, joita toteutetaan hajautetuilla kehitystiimeillä.

Keskisuurissa ja pienissä ohjelmistoprojekteissa päällekkäisyydellä voi olla iso vaikutus syntyvän jätteen määrään. Sommerville [10] sanoo tämän johtavan tilanteeseen, jossa aikaa kuluu enemmän siihen, miten ohjelmistoa tulisi kehittää, kuin kehitystyöhön ja testaamiseen itsessään. Sommervillen mielestä eräs syy perinteisen ohjelmistokehityksen ongelmiin on nykypäivän markkinoissa; ohjelmistoja kehitetään globaalissa, nopeasti muuttuvassa markkinatilanteessa, jossa yritysten on reagoitava nopeasti uusiin avautuviin mahdollisuuksiin markkinoilla. Tämä johtaa tilanteeseen, jossa vakaan ohjelmistoversion tuominen nopeasti markkinoille on mahdotonta. Vaatimukset voivat muuttua ennen projektin toimitusta ja työn aikana on mahdollista oppia uusia asioita. Joskus vaatimukset voivat yllättäen muuttua projektin toimituksen jälkeen suoritettujen asiakashaastattelujen perusteella. Uusista vaatimuksista seuraa projektin myöhästyminen, budjetin ylittyminen ja asiakkaan tyytymättömyys aikataulun myöhästymisen vuoksi [10].

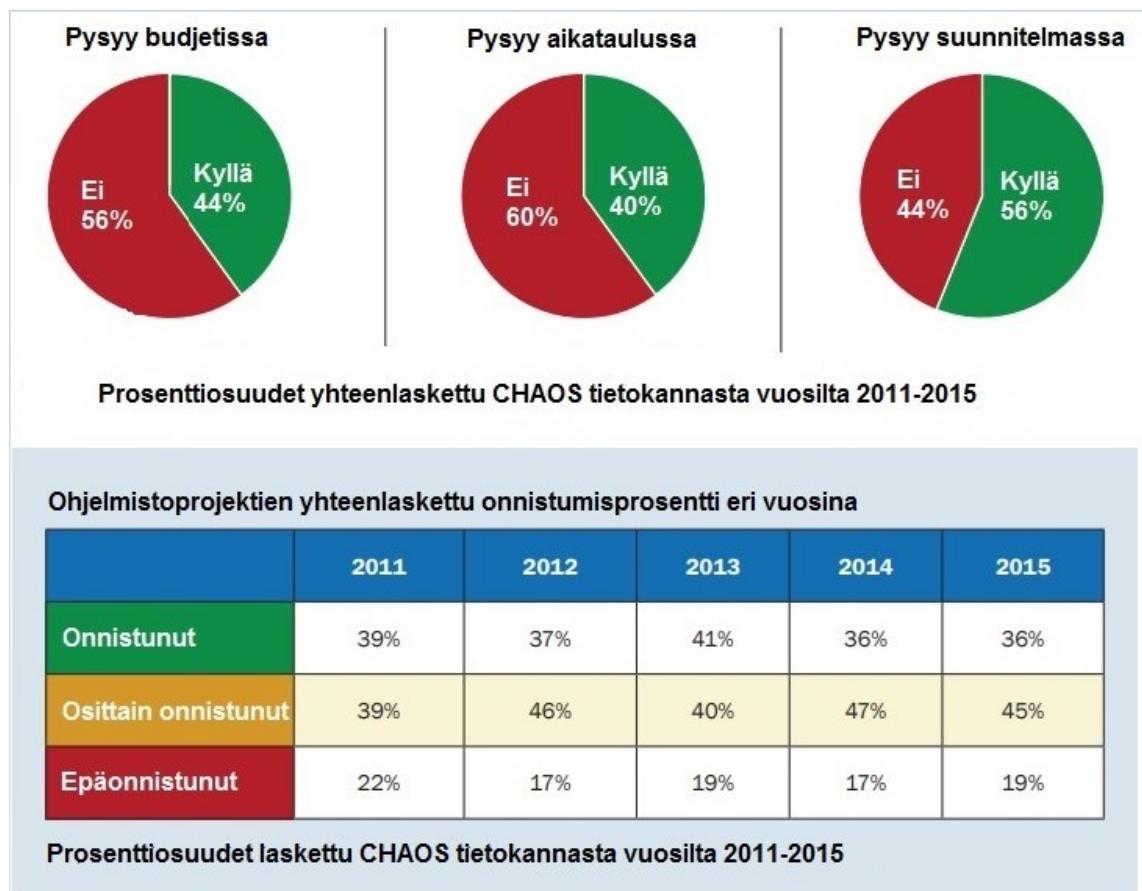
2.4 Tutkimuksia ohjelmistotuotannon haasteista

Standish Group on julkaissut satunnaisesti raportteja ohjelmistokehityksen haasteista ja ongelmista [3]. Eräs viimeisimmistä raporteista on julkaistu vuonna 2015 ja se perustuu vuosina 2011–2015 kerättyihin tietoihin. Ellei toisin mainita, seuraava tarkastelu perustuu Standish Groupin [3] vuoden 2015 CHAOS-raporttiin.

Standish Groupin [3] tutkimuksessa on kerätty tietoa kyselytutkimuksen avulla ohjelmistoprojekteista. Peruskysymys on se, onko ohjelmistoprojekti ollut onnistunut, onnistunut osittain vai epäonnistunut (kuva 3). Ohjelmistoprojektin onnistumista mitataan kolmella osa-alueella; onko projekti pysynyt alkuperäisessä budjetissa, alkuperäisessä

aikataulussa tai onnistunut tuottamaan asiakastyytyväisyyden. Raportissa pureudutaan ohjelmistoprojektin onnistumisen ja epäonnistumisen kannalta keskeisiin elementteihin.

Asiakastyytyväisyys on tärkeä onnistumisen kriteeri. Ohjelmistoprojekti voi pysyä budjetissa, aikataulussa ja toimittaa alkuperäisen suunnitelman mukaisen ratkaisun saavuttamatta kuitenkaan asiakastyytyväisyyttä.



Kuva 3. Tilasto ohjelmistoprojektien onnistumisesta vuosina 2011–2015 (muokattu lähteestä [3]).

Tutkimuksen mukaan ohjelmistoprojektien asiakkailla on useita ennako-odotuksia ja kehitysideoita ohjelmiston toiminnallisuuteen ja ominaisuuksiin liittyen. Asiakastyytyväisyys on kuitenkin ollut korkeampi silloin, kun projekti on toimittanut oleelliset toiminnallisuudet ja ominaisuudet ja jättänyt pois asiakkaan alun perin haluamia yksityiskohtaisia ominaisuuksia. Yksityiskohtaiset ominaisuudet lisäävät kustannuksia, lisäävät monimutkaisuutta ja tuottavat vain harvoin niitä kompensoivaa lisäarvoa.

Ohjelmistoprojektin koko on isoin yksittäinen onnistumiseen vaikuttava tekijä. Tutkimus osoittaa selvästi sen, että mitä isompi ohjelmistoprojekti on, sitä todennäköisemmin se epäonnistuu. Mitä nopeammin projekti saadaan tuotantokäyttöön, sitä nopeammin se alkaa myös tuottaa arvoa investoinnille. On järkevää jakaa iso projekti pieniin osiin, sillä pienet projektit onnistuvat tutkimuksen mukaan huomattavasti todennäköisemmin.

Tutkimuksessa ohjelmistoprojektit jaettiin ketterällä menetelmällä suoritettuihin (Agile), ja perinteisen vesiputousmallin mukaisesti suoritettuihin (Waterfall) projekteihin (kuva 4). Tämän lisäksi projektit luokiteltiin kooltaan isoihin, keskisuuriin ja pieniin projekteihin. Ketterän kehittämisen metodeilla oli tutkimuksen mukaan todennäköisintä onnistua. Mitä isompi projekti, sitä todennäköisemmin se onnistuu ketterällä metodilla. Toisaalta pienten projektien kohdalla ero ei ole erityisen suuri.

Ketterä kehittäminen vs vesiputousmallin kehittäminen				
Projektin koko	Projektimalli	Onnistunut	Osittain onnistunut	Epäonnistunut
Kaiken kokoiset projektit	Ketterä	39%	52%	9%
	Vesiputousmalli	11%	60%	29%
Isot projektit	Ketterä	18%	59%	23%
	Vesiputousmalli	3%	55%	42%
Keskikokoiset projektit	Ketterä	27%	62%	11%
	Vesiputousmalli	7%	68%	25%
Pienet projektit	Ketterä	58%	38%	4%
	Vesiputousmalli	44%	45%	11%

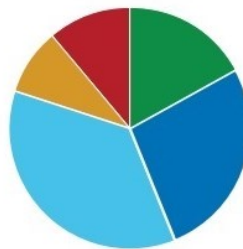
Prosenttiosuudet laskettu CHAOS tietokannasta vuosilta 2011-2015

Kuva 4. Ketterä versus vesiputousmalli jaettuna projektin koon mukaan 2011–2015 (muokattu lähteestä [3]).

Projektin koko tuo usein mukanaan monimutkaisuutta. Vaatimuksiltaan yksinkertaiset projektit taas onnistuvat todennäköisemmin kuin monimutkaiset projektit; tutkimuksessa erittäin monimutkaisista projekteista 15% luokiteltiin onnistuneiksi ja yksinkertaisista projekteista 38% onnistuneiksi. Ketteryydestä hyödytään siis enemmän suurissa projekteissa.

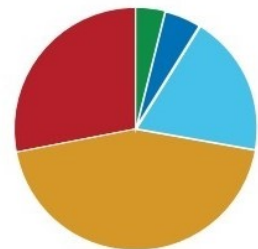
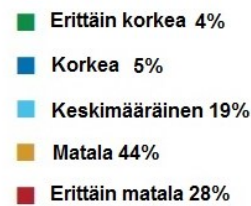
Onnistuneet ohjelmistoprojektit tarvitsevat viisaita ja koulutettuja ihmisiä, yksi tärkeimmistä onnistumiseen vaikuttavista tekijöistä ovat kokeneet ihmiset (kuva 5). Tutkimuksen mukaan kyvykäs, ketterä tiimi (Gifted Agile Teams) onnistuu projektin tavoitteissa huomattavasti todennäköisemmin kuin kokematon tiimi (Unskilled Agile Teams).

Kokeneet ketterät tiimit



Kokeneiden ketterien tiimien osaamispisteitys keskisuurissa ja suurissa ohjelmistoprojekteissa 2011-2015.

Kokemattomat ketterät tiimit



Kokemattomien ketterien tiimien osaamispisteitys keskisuurissa ja suurissa ohjelmistoprojekteissa 2011-2015.

Kyvykkyyden ja kokemuksen vaikutus projektin onnistumiseen

	Onnistunut	Osittain onnistunut	Epäonnistunut
Lahjakas	38%	45%	17%
Taitava	31%	53%	16%
Osaava	28%	53%	19%
Kyvykäs	24%	54%	22%
Ammattitaidoton	17%	60%	23%

Prosenttiosuudet laskettu CHAOS tietokannasta vuosilta 2011-2015

Kuva 5. Ketterien tiimien kyvykkyys ja projektien onnistuminen 2011 – 2015 (muokattu lähteestä [3]).

Isoissa projekteissa haasteena on löytää riittävästi kyvykkäitä resursseja. Tällöin joudutaan muodostamaan tiimi kyvykkäistä ja kokemattomista resursseista, koska kyvykkäitä resursseja on harvoin riittävästi tarjolla. Siksi isojen projektien resurssoinnissa joudutaan yleensä tekemään kompromisseja. Resursseja hankkiessa on tärkeää ymmärtää, minkälaisella osaamisella tiimiä tulee täydentää. Kyvykäs tiimi on yhdistelmä organisaation sisäisiä resursseja täydennettynä ostetuilla ulkoisilla resursseilla. Ohjelmistoprojektin koolla on siis iso vaikutus resurssointiin ja sitä kautta projektin onnistumiseen. Pienen projektin pieneen projektitiimiin on paljon helpompi saada kyvykkäät resurssit, kuin ison projektin isoon projektitiimiin.

3. SCRUM

Scrum on ketterä ohjelmistokehitysprosessi, jonka tarkoitus on lisätä tehokkuutta, selkeyttä, keskittymistä ja läpinäkyvyyttä ohjelmistonkehitystiimeihin ja projekteihin. Oikeaoppisesti käytöön otettu Scrum suunniteltiin nopeuttamaan projektin kehitystä, linjaamaan yksilölliset ja organisaatiotasoiset tavoitteet, luomaan organisaatiokulttuuri, joka perustuu tehokkuuteen, auttamaan sidosryhmiä arvon muodostamisessa, saavuttamaan tehokas ja jatkuva kommunikointi ja auttamaan yksilöitä kehittymään. Scrum tarjoaa siis kokonaisvaltaisen projektinhallintamenetelmän Agile Manifesto -arvomaailman pohjalta. [11].

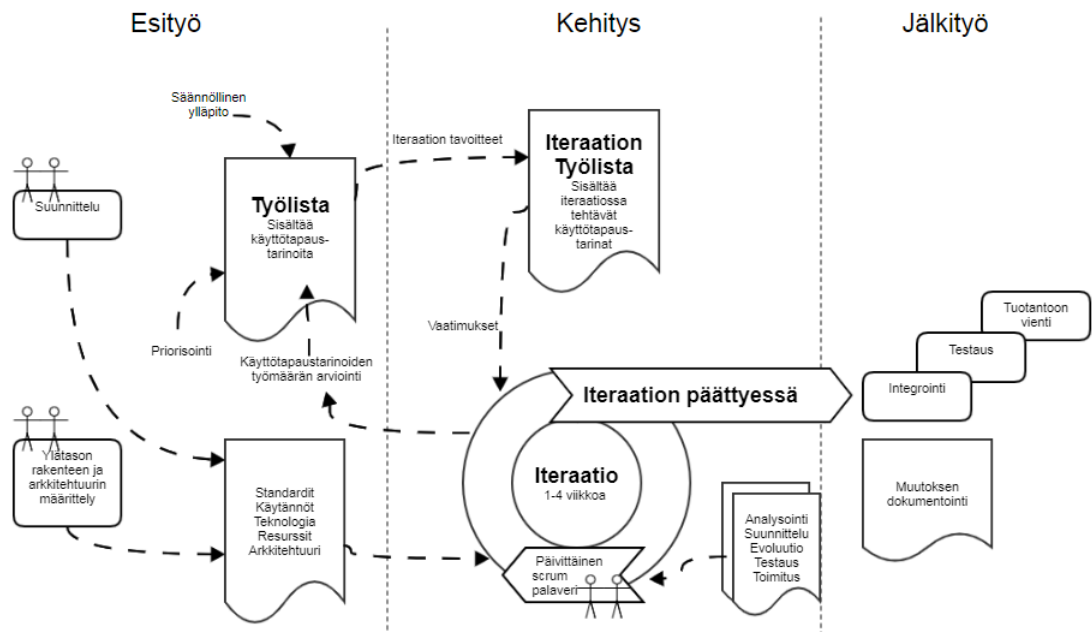
Scrumin käyttö ohjelmistokehitykseen alkoi Easel Corporationissa vuonna 1993 [11]. Käyttöön ottoon vaikuttivat seuraavat, perustavanlaatuiset ongelmat ohjelmistokehityksessä [11]:

- Epävarmuus on luontaista ja välttämätöntä ohjelmistokehityksessä.
- Kaikki vaatimukset ohjelmistolle eivät ole tiedossa, ennen kuin se on loppukäyttäjien käytössä.
- Käyttäjän kanssa vuorovaikutteisen ohjelmiston määrittely ennen aloittamista on hankalaa.
- Epäselvät ja muuttuvat vaatimukset yhdistettyinä muuttuviin työkaluihin ja teknologioihin tekevät käyttöönotosta vaikeasti ennustettavan.

Isojen ohjelmistohankkeiden hallitseminen on vaativa tehtävä ja riski ylittää suunniteltu aikataulu ja budjetti kasvaa projektin koon ja monimutkaisuuden mukana [12]. Leffingwell kuvailee Scrumin olevan kevyt ketterän projektimallin hallintametodi, joka perustuu pieniin itseohjautuviin tiimeihin, täydelliseen läpinäkyvyyteen, sekä nopeaan oppimis- ja mukautumiskykyyn [13]. Appelon puolestaan kutsuu Scrumia avoimeksi ketterän projektin viitekehyykseksi, määriteltynä prosessien tai metodologioiden sijaan [14].

3.1 Scrum-prosessi

Scrum-prosessiin kuuluu kolme vaihetta. Nämä vaiheet ovat suunnittelu (pregame), kehittäminen (development) ja lopuksi havainnointi ja kehittäminen (postgame) [15]. Tätä jakoa on havainnollistettu kuvassa 6.



Kuva 6. Scrum-prosessi (muokattu lähteestä [15]).

Usein siteeratun Schwaberin ja Beedlen mallin mukaisesti Scrum-prosessi toimii seuraavasti [15][16]:

- **Pregame** eli suunnitteluvaihe. Tämä vaihe koostuu kahdesta osavaiheesta; suunnittelu ja arkkitehtuurisuunnitelma/ylätason suunnitelma. Suunnitteluvaiheessa työestetään työlistaa (Product Backlog List). Työlista sisältää kaikki vaatimukset, jotka sillä hetkellä tunnetaan. Vaatimukset voivat tulla asiakkaalta, myyntiosastolta tai markkinointiosastolta, asiakastuesta tai ohjelmistonkehittäjiltä. Vaatimukset priorisoidaan ja niiden työmäärät arvioidaan. Työlistaa ylläpidetään jatkuvasti uusilla vaatimuksilla, joita priorisoidaan ja arvioidaan. Suunnitteluun kuuluu myös projektitiimin määrittely, tarvittavien työkalujen selvittäminen, ulkoisien resurssien ja riippuvuuksien määrittely, riskienhallinta, koulutustarpeet ja loppuhyväksyntäprosessi. Jokaisen

iteraation päätteeksi päivitetty työlista käydään projektitiimin kanssa läpi pyrkien sitoutumaan seuraavan iteraation tavoitteisiin.

Arkkitehtuurisuunnitelma tai toiminnallinen yleissuunnitelma tehdään työlistalla olevien vaatimusten perusteella. Mikäli kehitystä tehdään jo olemassa olevaan järjestelmään, arvioidaan vaatimusten aiheuttamat muutokset ja riskit olemassa olevaan järjestelmään. Suunnittelupalaverissa käydään läpi implementointiehdotukset ja arkkitehtuuripäätökset tämän perusteella. Alustava suunnitelma tulevasta ohjelmiston version julkaisusta voidaan myös tehdä tässä vaiheessa.

- **Development phase** eli iteraatio ja tekemisen vaihe. Kehittämisessä edetään Scrum-prosessin mukaisesti ja ollaan valmiita odottamattomiin muutoksiin. Ympäristön ja teknologian aiheuttamia odottamattomia muutoksia havainnoidaan ja hallitaan Scrum-prosessin eri vaiheissa. Tällaisia muuttujia voivat olla esimerkiksi muutos aikataulussa, vaatimuksissa, resursseissa, teknologioiden käyttöönotossa tai haasteet kehitysmenetelmissä. Sen sijaan että tällaisia muuttujia yritettäisiin tunnistaa ja vaikuttaa niihin ainoastaan projektin alussa, Scrumissa pyritään hallitsemaan muutoksia kehitysprosessin aikana ja ylläpitämään ketteryttä.

Kehitystyö tehdään iteraatioissa (sprint). Iteraation päätteeksi syntyy asiakasarvoa lisäävää toiminnallisuutta. Jokaiseen iteraatioon sisältyy perinteisen ohjelmistokehityksen elementtejä; vaatimuksien määrittelyä, analysointia, käyttöliittymän suunnittelua, elinkaaren suunnittelua ja tuotantojulkaisun suunnittelua. Iteraation kesto päätetään sen alussa. Kesto voi olla yhdestä viikosta kuukauteen. Vaikka jokainen iteraatio tuottaa arvoa tai toiminnallisuutta, voi ensimmäisen varsinaisen tuotantojulkaisun valmiiksi saaminen kestää esimerkiksi kolme tai kahdeksan iteraatiota. Ohjelmiston kehittämiseksi voi myös työskennellä useampi kehitystiimi samanaikaisesti omissa iteraatioissaan.

- **Postgame phase** eli kehitysjakson jälkeinen vaihe. Iteraatioon valitut vaatimukset ovat mahdollisesti tulleet valmiiksi ja hyväksytty julkaistavaksi. Tässä vaiheessa julkaisuvalmiit vaatimukset ovat testattu, dokumentoitu ja hyväksytty suoritetuiksi.

Yleinen käsitys on, että harvassa organisaatiossa Scrumin käyttöönotto tapahtuu täydellisesti teorian mukaisesti. Tutkimuksia Scrumin soveltamisesta organisaation haasteisiin tai syitä soveltamistarpeisiin ei ole tehty paljon. Michael Hron ja Nikolaus Obwegeserin tutkimuksen mukaan yleisimmät syyt muokata Scrum-prosessia ovat hajallaan oleva kehitystiimi, yhdistäminen muihin ohjelmistokehitysmetodeihin, lisääntyneet vaatimukset käytettävyyteen, Scrumin skaalaaminen horisontaalisesti

organisaatiossa, Scrumin skaalaaminen isompaan mittakaavaan, työkalurajoitteet tai valinnat, Scrumin soveltaminen erilaisesta näkökulmasta katsottuna [17].

3.2 Scrum-roolit

Schwaber ja Beedle määrittelevät seuraavat Scrum-roolit [15][16]:

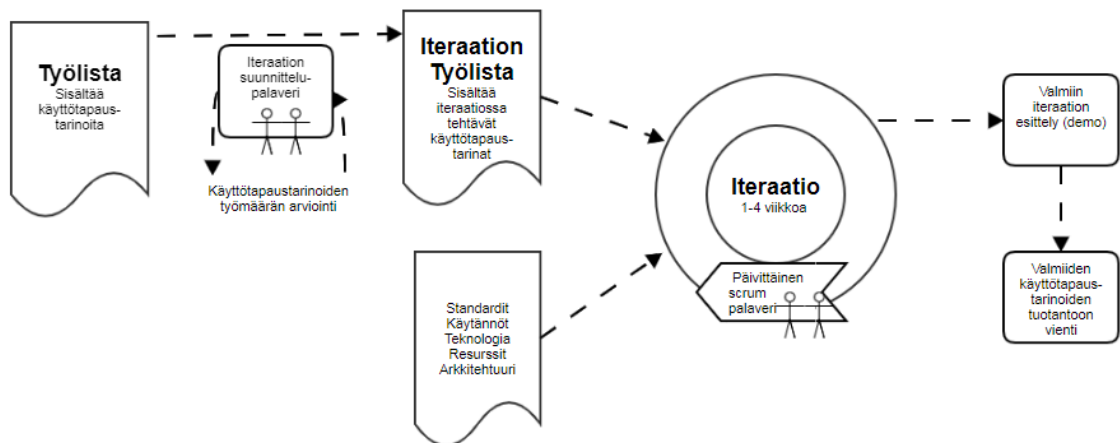
- **Scrum-mestari** (Scrum master) on hallinnollinen ja ohjaava rooli Scrumissa. Scrum-mestarin vastuulla on huolehtia projektin läpiviemisestä Scrumin ja ketterän kehittämisen käytäntöjen ja arvojen mukaisesti suunnitellussa aikataulussa. Scrum-mestari auttaa projektitiimin ja asiakkaan välisessä kommunikaatiossa ja vuorovaikutuksessa sekä huolehtii projektitiimin hallinnollisesta työstä. Scrum-mestarin vastuulla on myös poistaa mitä tahansa esteitä projektitiimin tieltä, jotta kehitystyö olisi mahdollisimman tehokasta.
- **Tuoteomistaja** (Product owner). Tuoteomistaja on virallisesti vastuussa projektista, sen hallinnoinnista, kontrolloinnista ja työlistasta (Backlog). Tuoteomistaja tekee päätökset työlistan priorisoinnista ja kehitykseen etenevistä vaatimuksista.
- **Scrum-tiimi** on itseohjautuva projektitiimi, joka suorittaa työlistan tehtävien arvioinnin ja tekemisen. Scrum-tiimi osallistuu iteraation vaatimusten valintaan ja ilmoittaa mahdollisista esteistä, jotka voisivat estää iteraation tavoitteeseen pääsemisen. Scrum-mestari osallistuu tuoteomistajan valintaan.
- **Asiakas** on osallisena vaatimusten määrittelyssä työlistalle. Yleisesti tuoteomistaja on asiakas.
- **Organisaation hallinto** (management) on vastuussa kokonaisuudessaan projektista. Hallinnon vastuulla on asettaa projektille tavoitteet. Hallinto valitsee tuoteomistajan ja valvoo projektin etenemistä.

Tuoteomistajan päätöksiä ja priorisointia ohjaa visio [18].

3.3 Scrum-prosessin tapahtumat

Projektinhallinnan työkalujen suhteen, Scrum ei määrittele tietyn työkalun käyttämistä. Sen sijaan Scrumissa vaaditaan tiettyjä hallinnollisia toimintoja ja niihin sopivien työkalujen käyttöä, jotta vältytään monimutkaisuuden ja ennakoimattomuuden aiheuttamalta kaaokselta. [16].

Kuten luvussa 2 mainittiin, Scrum on iteratiivinen viitekehys. Sovellusta kehitetään siis useassa iteraatiossa, kehityssyklissä. Yksittäinen Scrum-iteraatio ja sen rutiinit sekä toiminnot on kuvailtu kuvassa 7.



Kuva 7. Scrum-iteraatio (muokattu lähteestä [15]).

Scrum-prosessi koostuu seuraavista rutiineista ja toiminnoista [15][16]:

- **Työlista** (Backlog) sisältää kaikki sillä hetkellä tunnetut vaatimukset lopputuotteesta. Työlista elää jatkuvasti. Uusia tunnistettuja vaatimuksia tulee listalle, vaatimusten työmäärää arvioidaan ja priorisointi muuttaa listan järjestystä. Iteraation aluksi työlistalta valitaan vaatimukset, jotka halutaan saada valmiiksi seuraavan iteraation päätteeksi. Yksittäiset vaatimukset voivat olla toiminnallisuuksia, ominaisuuksia, teknologiapäivityksiä tai löydettyjä virheitä (bugs). Listalla voi olla myös vaatimus lisätutkimuksesta esimerkiksi teknologiaan liittyen, ennen kuin päätös teknologiasta voidaan tehdä. Vaatimuksia voi syntyä useista eri lähteistä. Tuoteomistaja on vastuussa työlistan ylläpitämisestä ja päivittämisestä.

- **Työmääräarviointi** (effort estimation). Scrum-tiimi suorittaa tuoteomistajan ohjaamana yksittäisten vaatimusten työmääräarvioinnin. Vaatimukset puretaan tarvittaessa pienempiin tehtäviin arvioinnin helpottamiseksi. Arviointi tarkentuu sitä mukaa, kun vaatimuksen sisältö tarkentuu.
- **Iteraatio** (sprint). Scrum-tiimi järjestäytyy itsenäisesti suorittamaan iteraatiota, joka kestää keskimäärin kolme kalenteriviikkoa. Iteraation tavoite on saada valitut vaatimukset valmiiksi. Scrum-tiimin työkaluja ovat iteraation suunnittelupalaverit (sprint planning meetings), työlistapalaverit (Sprint Backlog meetings) ja päivittäiset palaverit (Daily Scrum Meetings).
- **Iteraation suunnittelu** (sprint planning meeting) on Scrum-mestarin organisoima kaksivaiheinen kokous. Ensimmäisessä vaiheessa asiakkaat, tuoteomistaja, projektin hallinnolliset tahot ja Scrum-tiimi sopivat tavoitteet tulevalle iteraatiolle. Toisessa vaiheessa Scrum-tiimi ja Scrum-mestari suunnittelevat miten vaatimukset tulee toteuttaa iteraation aikana.
- **Iteraation työlista** (Sprint Backlog) on iteraatiossa suoritettavat vaatimukset, jotka tulivat valituiksi iteraation suunnittelupalaverissa. Iteraation työlista ei muutu iteraation aikana. Kun iteraation työlistaan valitut vaatimukset ovat tulleet suoritetuiksi, iteraatio on valmis julkaistavaksi.
- **Päivittäisten iteraatiopalaverien** (Daily Scrum Meetings) tarkoitus on seurata iteraation etenemistä. Palavereissa käydään läpi sitä, mitä on saatu aikaan viimekertaisen iteraatiopalaverin jälkeen ja mitä tehdään seuraavaksi. Samassa yhteydessä keskustellaan myös haasteista ja riskeistä, jotka hankaloittavat tai estävät iteraation tavoitteeseen pääsemistä. Iteraatiopalaveri pidetään iteraation jokaisena päivänä, se kestää noin 15 minuuttia ja siihen voi osallistua Scrum-tiimin lisäksi myös hallinnollisesta puolesta vastaavia henkilöitä.
- **Iteraation katselmointi** (Sprint Review Meeting) pidetään iteraation viimeisenä päivänä. Scrum-tiimi ja Scrum-mestari esittelevät aikaan saadut tulokset tuoteomistajalle, asiakkaille ja projektin hallinnollisille tahoille. Osallistujat tutustuvat valmiisiin ominaisuuksiin ja vievät ne eteenpäin käyttöönottoon. Katselmointi on myös hyvä hetki keskustella tulevasta iteraatiosta, työlistan prioriteeteista ja vaatimuksista.
- **Retrospektiivi** (Retrospective meeting) on iteraation jälkeinen Scrum-tiimin palaveri, jossa keskustellaan päättäneestä iteraatiosta; mikä meni hyvin ja mitä voisi parantaa, jotta työstä tulisi tuottavampaa ja miellyttävämpää.

3.4 Scrum-kommunikaatio

On yleisesti tiedostettu, että huono kommunikaatio on eräs merkittävin syy ohjelmistoprojektin epäonnistumiselle. Huono kommunikaatio johtaa helposti oletuksiin, väärinymmärryksiin ja luottamuspulaan [19].

Tehokas kommunikointi vaatii useita kokouksia. Osa ihmisistä kokee kokoukset keskittymistä häiritseväksi ja työmäärää lisääväksi. Niinpä osa kehitystiimin jäsenistä saattaa suhtautua välinpitämättömästi kokouksiin, joita he eivät koe kiinnostavaksi [20].

Kyselytutkimuksessa 'Issues and challenges in Scrum implementation' todetaan, että työskentely usean kehitystiimin kanssa on hankala tehtävä ja Scrumista ei löydy selkeää ohjetta tämän tilanteen hallitsemiseen. Scrumista löytyvä metodi 'Scrum of Scrums' ei ole erityisen toimiva tilanteissa, joissa kehitystiimit on jaettu työskentelemään omissa erillisissä osa-alueissaan. Tilanne on erityisen vaativa Scrum-mestarille, jolle tiimien koordinointi aiheuttaa lisätyötä. Agile ja Scrum – projektikehyksestä yksinkertaisesti puuttuu ohje prosessiin, jolla olisi mahdollista hallita yhtä aikaa useampaa kehitystiimiä [20].

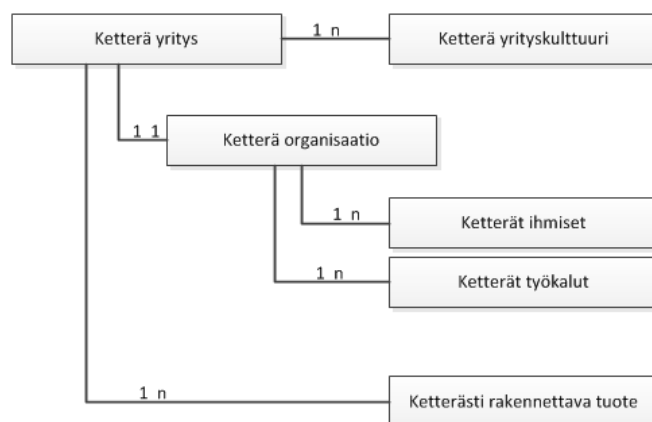
Eräs tapaustutkimus osoittaa, kuinka hajautettu, ja jopa ulkoistettu kehitystiimi kykenee olemaan yhtä tehokas kuin yhteen koottu kehitystiimi [11]. Tässä onnistuminen vaatii erinomaista Scrumin käyttöönottoa, sekä hyviä ohjelmistokehitysmenetelmiä. Koko joukon pienempiä tiimejä pitää kyetä toiminaan yhtenä tiiminä ja etenemistä pitää hallita yhdessä yhteisessä paikassa. Projektin seurantatyökalut, raportointi ja päivittäiset kokoukset täytyy niin ikään tehdä yhdessä.

3.5 Scrum isossa organisaatiossa

Ketterät menetelmät, esimerkiksi Scrum, on alun perin suunniteltu toimimaan pienissä projektitiimeissä, joissa projektin jäsenet työskentelevät samassa toimistossa ja samassa työhuoneessa. Nykyisin useat yritykset hyödyntävät ketteriä menetelmiä myös suurissa ohjelmistoprojekteissa, joissa useat projektitiimit työskentelevät

maantieteellisesti useammassa toimipaikassa. Tavoitteena on skaalata ketterien menetelmien hyödyt isompaan mittakaavaan. Skaalaamisesta seuraa kuitenkin haasteita, esimerkiksi tiimien välinen koordinointi, vaatimusten analysointihaasteet ja tiimien vastuiden epäselvyys [21]. Alkuperäisessä muodossaan Scrum, ja sen työkalut on suunniteltu toimimaan organisaation ruohonjuuritasolla. Ohjelmistokehityksen elinkaari yhdistettynä strategiaan ei siis ole osa Scrumia [22]. Leffingwellin mukaan [13] tehokkaan ketterän mallin implementointi yrityksessä ei ole pieni tehtävä. Jopa kyvykkäillä yrityksillä tai pienemmillä organisaatioilla saattaa kulua puolesta vuodesta vuoteen saada ketterän kehittämisen peruselementit hallintaan. Parhaan tuottavuuden ja laadun saavuttaminen, sekä merkittävän yrityskulttuurin muutoksen saavuttaminen vie useampia vuosia.

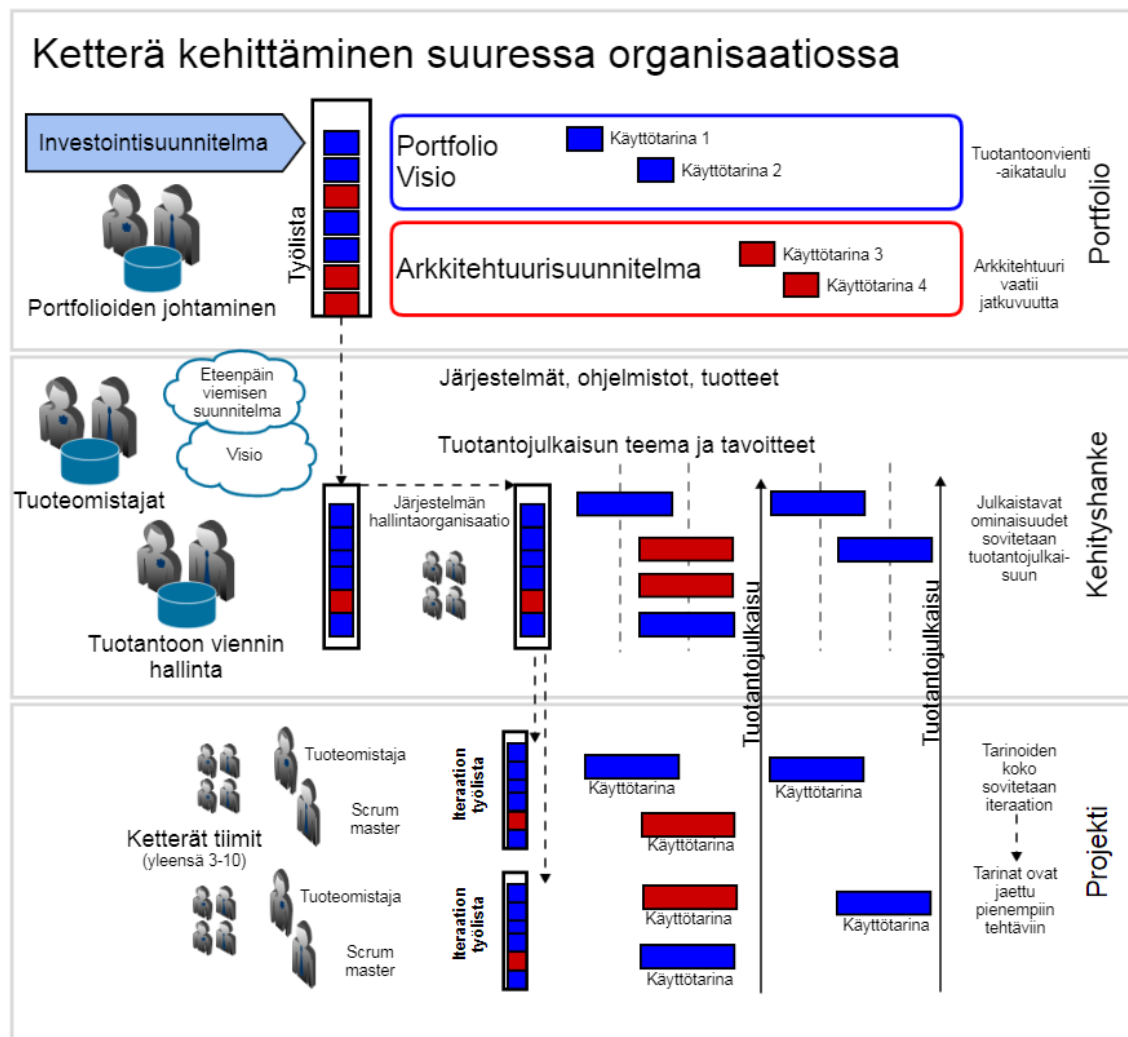
Petri Kettusen ja Maarit Laantin mukaan on useita tapoja implementoida ketterä toimintamalli yritykseen hyödyntäen joustavaa organisaatiomallia [23], Scrumista julkaistu kirjallisuus kannustaa organisaatiossa limittäin toimiviin, itse ohjautuviin Scrum-tiimeihin. Joustavuutta voidaan lisätä sillä, että Scrum-tiimiä ei rajoiteta toimimaan tietyn arkkitehtuurin tai teknologian alueella. Organisaatiota voidaan edelleen pitää ketteränä, vaikka tällaisia vapauksia ei sallittaisikaan. Yrityksen prosessit, työskentelytavat, säännöt ja käytännöt määrittelevät sen, miten paljon joustavuutta on mahdollista sallia. Jos ketterät menetelmät otetaan käyttöön vain yhden tiimin tasolla, hyödyt nähdään myös vain tiimitasolla. Jotta hyödyt voidaan nähdä koko yrityksen tasolla, pitää ketterät toimintatavat ottaa käyttöön ja ymmärtää koko organisaatiossa. Kettunen & Laanti kuvailee ketterän yrityksen konseptia kuvassa 8.



Kuva 8. Kettusen ja Laantin ketterän yrityksen konsepti (muokattu lähteestä [23]).

Skaalattaessa ketterä menetelmä isoon organisaatioon, useaan itsenäiseen tiimiin ja useaan eri lokaatioon, on erityisesti huomiotava ihmiset, vuorovaikuttaminen ja kommunikointi [24]. Ecksteinin mukaan ihmisten kyky ottaa vastuuta pitää huomioida tällaisessa tilanteessa aivan erityisesti. Ketterän kehityksen arvot ovat tärkeässä roolissa; tiimin tuottavuus on tärkeämpää kuin yksilön tuottavuus. Tämä tarkoittaa sitä, että tiimin onnistuessa tehtävässään, myös yksilö onnistuu. Onnistumisen kannalta yksilöiden kyky ottaa vastuuta, luottamus ja tiimityöskentelykyky ovat tärkeämpiä kuin tehtävien käskyttäminen ja kontrollointi. Johtamisstrategian tulee perustua luottamukseen [24].

Dean Leffingwell menee organisaatioanalyysissä Laantia ja Ecksteiniä pidemmälle yksityiskohdissa. Leffingwell kuvaa ketterää organisaatiomallia, sen prosesseja ja henkilörooleja yksityiskohtaisesti ”The big picture of enterprise agility” -kirjansa Whitepaper-osassa [13]. Leffingwellin prosessimalli ketterästä organisaatiosta kuvassa 9.



Kuva 9. Leffingwellin ketterän yrityksen konsepti (muokattu lähteestä [13]).

Leffengwell [13] on jakanut kokonaisuuden kolmelle tasolle. Näitä ovat projektitaso (Project level), ohjelmataso (Program level) ja portfoliotaso (Portfolio level). Seuraavassa näitä tasoja ja niiden liittymistä toisiinsa tarkastellaan tarkemmin.

Projektitasolta löytyvät kehitystiimit. Kehitystiimejä voi olla yksi tai useampi. Tiimissä toimii mieluiten 3-5 kehittäjää. Tiimissä on kaksi määriteltyä roolia, tuoteomistaja (Product owner) ja kehitystä ohjaava rooli (Agile master), joka Scrumissa tunnetaan Scrum-mestarin roolina. Koska kehitystiimi on vastuussa tarinoiden määrittelystä, tekemisestä ja testaamisesta, on hyödyllistä täydentää tiimiä testaamisesta vastaavalla henkilöllä. Tehokkaan tiimin koko on enimmillään 7-9 henkilöä. Suurissa yrityksissä kehitystiimit tekevät yhteistyötä kehittäessään yhteistä lopputulosta. Kehitystiimit

työskentelevät omien vaatimustarinoidensa kanssa (Story) ja tiimien yhteistyössä syntyy arvoa tuottava lopputuote tai arvoketju (Value stream) [13].

Ohjelmatasolla hallitaan isoja kokonaisuuksia. Isot kokonaisuudet muodostuvat useasta projektitason kehitystiimistä ja niiden tuottamasta arvoketjusta. Arvoketjujen lopputulokset yhdistetään tuotantoversioiksi ”Agile Release Train” [13].

Portfoliotasolla hallitaan kokonaisuuden visiota ja arkkitehtuurivalintoja. Visiot syntyvät yrityksen strategiasta ja sijoituspäätöksistä (Investment theme). Arkkitehtuurivalinnat pitäisi lähtökohtaisesti tehdä tiimitasolla, sillä ketterät kehitystiimit ovat itsejohtautuvia (self-organizing teams). Yritys-, tai konsernitason on usein kuitenkin järkevää rajoittaa arkkitehtuuriset valinnat yrityksen kokonaisuuden kannalta järkevällä tavalla [13].

Koko prosessin tavoitteena on julkaista ohjelmiston versiopäivityksiä loppuasiakkaalle. Isossa yrityksessä kokonaisuus on usein monimutkainen prosessi. Tämän vuoksi Leffingwell tunnistaa vielä kaksi tiimiä: järjestelmätiimi (System team) ja julkaisuprosessista vastaava tiimi (Release management team). Nämä kaksi tiimiä vastaavat usein järjestelmätason regressiotestauksesta ja hyväksyvät version asiakkaalle julkaisemisen ja vastaavat myös julkaisuprosessista [13].

Leffingwellin [13] mukaan työtehtävien tittelit voivat vaihdella yrityskohtaisesti, mutta on erittäin tärkeää ymmärtää erilaiset roolit, sillä niiden avulla määritellään, kuka on vastuussa mistäkin osa-alueesta. Pienemmissä organisaatioissa sama henkilö saattaa hoitaa useampaa roolia.

4. LÄHTÖKOHTA JA TUTKIMUSPROSESSI

4.1 Tutkimuksen lähtökohta

Tämän tutkimuksen tarkoituksena on kartoittaa, kuinka hyvin tutkimustapauksen yrityksessä Scrum-metodologialla toimiva pilottiprojekti on onnistunut kommunikoimaan projektia seuraaville tahoille projektin etenemisestä. Tässä yhteydessä kutsumme tätä läpinäkyvyydeksi. Samassa yhteydessä pyrimme selvittämään näiden ulkopuolisten tahojen tietotaidon ketterään kehittämiseen (Agile Mindset). Haastateltavan tietotaito ketteryydestä oletettavasti vaikuttaa odotusarvoon ketterän projektin seuraamisesta.

4.2 Tutkimuksen kohdeyritys ja sen organisaatio

Kohdeyrityksellä on pitkä historia hyödyntää sähköisiä palveluita tuotteidensa tukena. Varaosien verkkokauppa on ollut käytössä 2000-luvun alusta alkaen ja pian tämän jälkeen varaosakaupankäyntiä on ollut mahdollista käydä myös suorien XML-pohjaisten integraatioiden kautta. Iso askel sähköisten palveluiden portfolioon tehtiin vuonna 2014 julkaisemalla asiakasportaali, jossa varaosakaupan lisäksi oli tarjolla muun muassa takuu- ja huoltopyyntöilmoitukset, huoltokirjeet ja varaosakirjat. Pian tämän jälkeen portaali tarjosi asiakkaalle mahdollisuuden ottaa yhteyttä kohdeyritykseen myös chat-keskustelun kautta. Monessa osassa rakennetun portaalin ylläpito ja tekninen velka alkoi kuitenkin kasvaa liian suureksi. Portaali ei skaalautunut mobiililaitteen pienelle näytölle ja etenkin käyttöliittymän uudelleen suunnittelulle oli asiakaspalautteen mukaan tarvetta. Tämän seurauksena keväällä 2018 käynnistettiin kohdeyrityksessä asiakasportaalin uudelleenrakentamisprojekti.

Projekti on jakautunut kahteen kehitystiimiin (Kuva 10). Tiimit ovat varaosakauppaa rakentava tiimi sekä muita palveluja rakentava tiimi. Asiakasportaalitiimin vastuulla on portaalialusta, käyttökokemus kokonaisuudessaan ja kaikki muut portaalin sähköiset palvelut paitsi varaosakauppa. Teknologisesti portaalin käyttöliittymä on rakenteilla pilvipalveluun ja sitä tuetaan API-rajapinnan kautta.

Asiakasportaalitiimi koostuu seuraavista rooleista; tuoteomistaja, Scrum-mestari, kaksi palvelua ja käyttöliittymää suunnittelevaa henkilöä, neljä käyttöliittymää tekevää henkilöä, ja kaksi integraation asiakashallintajärjestelmän kanssa työtä tekevää

henkilöä. Lisäksi asiakasportaalitiimiin kuuluu viisihenkinen sähköisten palveluiden asiakastuki, joka on myös osa kehitystiimiä ja vastuussa testauksesta. Lisäksi tiimiä tukee hetkellisesti useita muita henkilöitä, jotka ovat vastuussa API-palvelun kautta kytketyistä tietojärjestelmistä. Fyysisesti portaalitiimi on jakautunut viiteen eri toimipaikkaan.

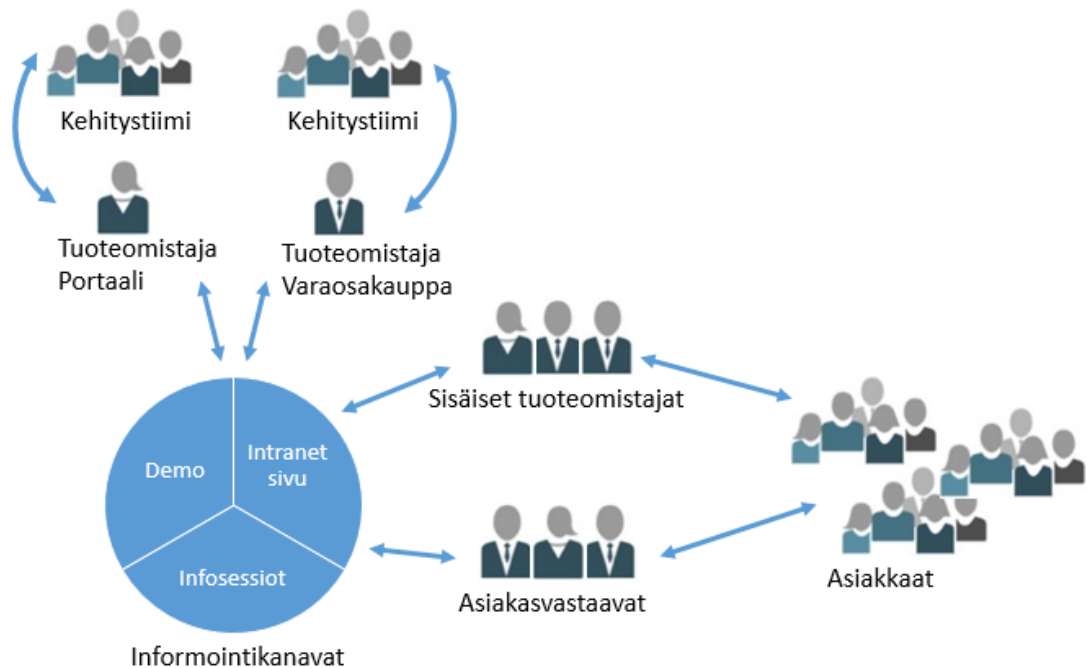
Varaosakauppatiimi koostuu seuraavista rooleista; tuoteomistaja, Scrum-mestari, palvelusuunnittelija ja käyttöliittymäsuunnittelija, kaksi käyttöliittymää tekevää henkilöä ja yksi integraation asiakashallintajärjestelmän kanssa työtä tekevä henkilö. Lisäksi tiimiä tukee hetkellisesti useampi toiminnanohjausjärjestelmän integraatiosta vastaava henkilö.

Tuoteomistajat toimivat projektin informoimisessa yhdessä. Molempien projektien iteraatioiden pituus on kaksi viikkoa, ja iteraatiot alkavat ja loppuvat yhtä aikaa. Myös iteraatioista tiedottaminen tapahtuu samoissa kommunikointikanavissa yhtä aikaisesti. Iteraatioita oli tutkimushetkellä suoritettu yhteensä 26 kertaa.

Iteraation viimeisenä päivänä järjestetään demo-tapahtuma, jossa kerrotaan iteraation aikana valmiiksi saadut käyttäjäkertomukset. Samassa yhteydessä kehitystiimit pitävät usein myös käyttöliittymän prototyypin esittelyn, jossa esitellään tulevien iteraatioiden mahdollisia ratkaisuja loppukäyttäjän näkökulmasta. Demo-tapahtumat ovat tärkeä osa kommunikointia sisäisille omistajille ja asiakasvastaaville siitä, mitä projektissa on tähän mennessä saatu valmiiksi, ja mitä tuleva iteraatio mahdollisesti sisältää. Valtaosa osallistujista osallistuu demo-tapahtumaan webinaarin kautta.

Infosessio-tapahtumat järjestetään demon jälkeen webinaarina. Niiden tarkoitus on sama kuin demossa, mutta muoto on tiiviimpi. Toisen infosession kohderyhmä ovat erityisesti yrityksen Aasiassa sijaitsevat asiakasvastaavat, ja toisen kohderyhmää ovat Pohjois- ja Etelä-Amerikassa sijaitsevat asiakasvastaavat.

Kolmantena viestintäkanavana toimii yrityksen intranet, jossa kerrotaan iteraatiossa valmiiksi saadut käyttäjätarinat. Projektilla oli intranetissä oma sivusto projektiviestintää varten. Intranetistä löytyivät valmiiden iteraatioiden lisäksi tiedot projektin missiosta ja visiosta.



Kuva 10. Tapausprojektin rakenne.

Projekti kommunikointia oli järjestetty koordinoitusti seuraavilla tavoilla;

- **Demo.** Iteraation esittely (Sprint demo) Skypen välityksellä iteran viimeisenä päivänä. Demo oli enintään tunnin mittainen. Demossa esiteltiin aluksi iteratiossa valmistuneet tarinat (Story) sekä esiteltiin tarinat ruutujakoa käyttäen. Tämän jälkeen demo-tilaisuuteen osallistuvilla oli mahdollisuus esittää avoimia kysymyksiä iteratioon tai itse projektiin liittyen. Useimpina kertoina demossa esiteltiin myös analyysivaiheessa olevien tarinoiden käyttöliittymän prototyyppiä. Myös prototyypistä oli mahdollisuus käydä avointa keskustelua ja tarinoita myös muutettiin saadun palautteen mukaan.
- **Infosessiot.** Demon jälkeisenä viikkona järjestettiin niin ikään Skypen välityksellä infosessioksi kutsuttu tilaisuus. Infosessioissa aika oli rajoitettu 30 minuuttiin. Tässä kokouksessa kerrottiin lyhyesti valmistuneet tarinat ja esiteltiin tarinat soveltuvin osin. Aikataulun sallimissa puitteissa oli myös mahdollisuus esittää kysymyksiä ja kommentteja.
- **Intranet.** Yammer-ryhmässä julkaistiin iteran päätyttyä valmistuneet tarinat. Yammer-ryhmässä oli myös mahdollisuus käydä keskustelua ja esittää kysymyksiä projektitiimille. Yammerin käyttäjät saavat myös sähköpostin kaikista uusista Yammer-viesteistä asetuksiaan muuttamalla. Sähköpostissa käsiteltiin epäsäännöllisesti projektin edistymistä sekä suurimpien välivaiheiden

saavuttamista. Sähköpostia ei käytetty iteraatioiden esittelyyn. Intranetissä julkaistiin satunnaisesti uutisia projektiin liittyvien välivaiheiden saavuttamisesta.

4.3 Menetelmät

Tämä tutkimus suoritetaan tapaustutkimuksen menetelmänä ja tutkimustietoa kerätään haastattelujen avulla. Haastatteluiden avulla pyritään saamaan laadullista tietoa tutkimuskysymykseen [25]. Kohdeorganisaatio haluaa kehittyä ohjelmistoprojektien tekemisessä ketterämpään suuntaan. Tapaustutkimuksen projekti on eräs ensimmäisistä ketterän mallin projekteista kohdeorganisaatiossa, ja ketterä Scrum-malli toimii stage-gate viitekehyksen sisällä. Tutkimus on kiinnostava, koska Scrum-mallin käyttöönotto sellaisenaan on suuressa organisaatiossa haasteellista. Koska tutkimuskysymyksenä on projektin läpinäkyvyys, haastateltavat ovat projektitiimin ulkopuolella olevia tahoja. Organisaatio toimii useassa eri maassa, toimipaikassa ja kulttuurissa. Tämä on haaste kommunikoinnille. Tapaustutkimuksen pilottiprojektin myötä avautuu mahdollisuus tutkia projektista kiinnostuneita tahoja ja kehittää ketterän kehittämisen viestintää kohdeorganisaatiossa.

Haastateltavat valittiin etukäteen tutkimuksen empiiriseen osuuteen. He ovat tutkimuksen kannalta sopiviksi katsottuja tahoja, jotka toimivat eri puolilla organisaatiota ja hyvin erilaisissa tehtävissä. Jokaisella haastateltavalla on jonkinlainen intressi tapausprojektiin. Valinta haastateltaviksi valittiin yhdessä keskustellen tuoteomistajan ja tutkimusavustajien kanssa. Haastattelut suoritettiin yksitellen skype-puheluna. Haastattelutilaisuus oli puolistrukturoitu. Haastattelukysymykset käytiin läpi tilaisuuden aikana rajoittamatta keskustelua, mikäli keskustelu etenee tutkimuksen kannalta mielenkiintoiseen suuntaan. Haastattelut kestivät 30–60 minuuttia.

Kaikki haastattelut tehtiin Skype-puhelua käyttäen 13.5.2019 – 3.6.2019 välisenä aikana. Haastattelun aikana pidettiin esillä haastattelun muistilistaa (Liite B ja Liite C). Tämä loi jokaiseen haastatteluun selkeän rakenteen rajoittamatta keskustelua kuitenkaan liikaa. Haastattelut alkoivat haastateltavan taustaa kartoittavilla kysymyksillä. Aloituskysymykset koskivat haastateltavan asemaa organisaatiossa, ketterän kehittämisen koulutustaustaa sekä käytännön kokemuksia ketteristä ohjelmistoprojekteista. Taustan kartoittaminen on tässä yhteydessä tärkeää, jotta on helpompi hahmottaa näkökulma, josta haastateltava kysymyksiin vastaa. Haastatteluista kaksi käytiin englannin kielellä ja viisi suomen kielellä.

Taustan kartoittamisen jälkeen haastateltavaa pyydettiin omin sanoin kertomaan tapausprojektin kulku alkupisteestä nykyhetkeen. Tällä kysymyksellä oli kaksi tarkoitusta; virittää haastateltava aiheeseen ennen varsinaisia tutkimuskohteeseen liittyviä kysymyksiä sekä hahmottaa, kuinka intensiivisesti haastateltava on seurannut projektin kulkua. Pitkä ja yksityiskohtainen kertomus projektin kulusta kertoo oletettavasti intensiivisestä projektin seuraamisesta. Lyhyehkön selostuksen voidaan olettaa johtuvan projektin pintapuolisemmasta seuraamisesta.

Tämän jälkeen haastattelussa siirryttiin itse tutkimuskysymyksiin. Tutkimuskysymykset käytiin läpi myös silloin, vaikka vastaus kysymykseen olisi tullut jo aiemmissa vastauksissa.

- **Miten olet osallistunut projektin seuraamiseen?**

Tämän kysymyksen tarkoituksena oli selvittää haastateltavan käyttämiä viestintäkanavia. Oletuksena oli kuulla projektissa käytetyistä informointikanavista, joita on esitelty aiemmin kuvassa 10. Samalla saatiin kuva, kuinka aktiivisesti haastateltava seurasi projektia.

- **Oletko saanut tietoa sinulle tärkeistä asioista?**

Tällä kysymyksellä oli kaksi tarkoitusta. Ensimmäinen tarkoitus oli se, onko projektin viestintä tavoittanut haastateltavan henkilön riittävällä tasolla. Tämän lisäksi kysymys avaa mahdollisuuden keskusteluun siitä, mitä asioita haastateltava pitää tärkeinä. Oletuksena oli, että henkilöt eri puolilta yrityksen organisaatiota ovat kiinnostuneita eri asioista ja seuraavat projektia erilaisista näkökulmista. Projektin viestintä pyrittiin suorittamaan riittävän laadukkaasti ja tehokkaasti. Mutta onnistuttiinko siinä?

- **Oletko kyennyt saamaan kuvan projektin saavutuksista tähän mennessä?**

Tämän kysymyksen tarkoitus oli selvittää, onko haastateltava henkilö saanut riittävästi kuvaa projektin etenemisestä ja saavutetuista etapeista (milestones). Tällä kysymyksellä voidaan myös kartoittaa haastateltavan käsitystä projektin visiosta ja missiosta.

- **Oletko kyennyt saaman kuvan projektin tavoitteista?**

Tämän kysymyksen tarkoitus oli selvittää, ovatko haastateltavat tietoisia siitä, miksi projektia ylipäättänsä tehdään. Myös tässä kysymyksessä kartoitetaan haastateltavan käsitystä projektin visiosta ja missiosta.

- **Oletko kokenut kykeneväsi vaikuttamaan projektin lopputulokseen?**

Tämän kysymyksen tarkoitus oli selvittää, ovatko projektia ympäröivä organisaatio ja tahot kokeneet voivansa riittävällä tavalla vaikuttaa projektin sisältöön ja yksityiskohtiin.

Haastattelut käytiin puolistukturoituina keskusteluina seuraten haastattelun kysymyslistaa. Tutkimustietoa voidaan kerätä kvalitatiivisesti sekä kvantitatiivisesti. Kvantitatiivisesti voimme tarkastella kuinka monta osallistujaa tapausprojektin demokokouksiin, infosessioihin osallistui, sekä kuinka moni on kirjautunut projektin Yammer-ryhmään. Itse haastattelut tuottavat kvalitatiivista tutkimustietoa. Haastatteluissa voidaan selvittää henkilöiden nykyhetkistä tyytyväisyyttä ja mielialaa. Haastatteluissa voidaan myös keskustella mahdollisista parannusehdotuksista.

4.4 Tiedon analysointi

Haastattelut nauhoitettiin ja kirjoitettiin nauhoituksen avulla auki. Tutkimusta kirjoittaessa litteroituja haastatteluja analysoitiin, tapaututkimuksen vastauksia ryhmiteltiin ja mielenkiintoisiksi katsottuja viittauksia käytettiin hyväksi.

Tutkimusdatan analysointiin vaikutti omalta osaltaan se, että tutkimuksen tekijä osallistui tapausprojektiin Scrum-mestarin roolissa. Tutkimuksen tekijällä on myös pitkä historia ja kokemus ohjelmistokehityshankkeissa kohdeorganisaatiossa. Tällä kokemuksella on omanlainen vaikutus valituiksi tulleisiin lainauksiin. Tämä tausta on myös vaikuttanut tutkimusdatan analysointiin.

Osa haastatteluista tehtiin englannin kielellä käyttäen englanninkielistä kysymyslistaa, Liite C. Suomenkielisten haastattelujen apuna taas käytettiin suomenkielistä kysymyslistaa, Liite B.

4.5 Tutkimuksen luotettavuus ja yleistettävyys

Haastateltavilta kysyttiin projektin kokemuksia ja mielipiteitä yhdellä hetkellä. Haastattelut suoritettiin, kun projektia oli tehty noin yksi vuosi. Tämän hetken jälkeen projektin viestintä on voinut muuttua. Myös haastateltavan intressi, mielenkiinto ja aktiivisuus projektia kohtaan on voinut muuttua haastattelun jälkeen. Haastatelluista henkilöistä useampi on myös vaihtanut työtehtäviä haastattelun jälkeen.

Haastateltaviksi valittiin henkilöitä projektin tiimistä, eri asianomistajia sekä muualta organisaatiosta sopiviksi katsottuja henkilöitä. Haastateltavia oli yhteensä seitsemän. Haastattelujen tulos jakautui tyytyväisiin henkilöihin, osittain tyytyväisiin henkilöihin ja tyytymättömiin henkilöihin.

Haastateltavien mielikuvaan ja ajatukseen voi vaikuttaa yleinen mielikuva itse projektista, sen tavoitteista tai henkilöistä. Haastateltavat eivät välttämättä kykene tarkastelemaan projektin viestintää pelkästään objektiivisesti. Myös ketterän kehittämisen koulutustausta, sekä projektikokemus ovat voineet vaikuttaa vastauksiin. Haastatelluista löytyi sekä ketterän kehittämisen koulutustaustan omaavia, että ensimmäistä kertaa ketterää projektia seuraavia henkilöitä.

Mikäli haastattelujen määrää olisi lisätty, olisi tulokset oletettavasti edelleen jakautuneet tyytymättömiin, osittain tyytyväisiin ja tyytyväisiin henkilöihin. Lisähaastatteluilla olisi kuitenkin voinut löytyä uusia näkökulmia projektiviestintään. Tutkimusotannon voidaan kuitenkin katsoa olevan määrällisesti edustava, sillä kuudennessa ja seitsemännessä haastattelussa ei tullut samalla tavalla uusia asioita ilmi kuin aiemmissa haastatteluissa.

Tutkimuksessa käytettiin hyväksi myös kvantitatiivista dataa. Haastattelujen yhteydessä tarkasteltiin aktiivisuutta eri viestintäkanavissa. Viestintäkanavien aktiivisuus kirjattiin iteraation 26 esittelyn kohdalla, tai sen jälkeen. Tuloksen voidaan katsoa edustavan juuri sen hetkistä tilannetta.

5. KYSELYTUTKIMUKSEN TULOKSET

Tutkimuksen haastateltavaksi valittiin sopiviksi katsottuja henkilöitä pitäen otanta mahdollisimman edustavana. Haastateltavissa oli niin projektissa työskenteleviä henkilöitä, kuin myös projektia satunnaisesti seuraavia henkilöitä.

- Henkilö 1 toimii tietohallinnossa, vastuualueenaan ketterä kehittäminen kohdeyrityksessä.
- Henkilö 2 toimii myös tietohallinnossa, vastuualueenaan tietojärjestelmäkehityshankkeet yrityksen erään divisioonan kanssa. Hän toimii lisäksi muun muassa projektien ohjausryhmissä.
- Henkilö 3 työskentelee globaalin myyntitiimin vetäjänä.
- Henkilö 4 toimii ryhmänvetäjänä kohdeyrityksen asiakastuessa. Henkilö on toiminut myös asianomistajana tapausprojektissa.
- Henkilö 5 toimii arkkitehtina ja ohjelmistokehittäjänä tutkimuksen kohteena olevassa projektissa. Henkilö 5 ei ollut yrityksen työntekijä, vaan hän työskenteli projektissa konsulttina.
- Henkilö 6 toimii tapausprojektin yhtenä tuoteomistajista, projektipäällikkönä ja asianomistajana.
- Henkilö 7 työskentelee myynnin tukihenkilönä eräässä myyntiorganisaatiossa.

Haastattelujen kulku oli jaettu neljään aihekokonaisuuteen; haastateltavan tausta suhteessa ketterään kehittämiseen, kertomus omin sanoin projektin kulusta, itse tapaus tutkimukseen liittyvät kysymykset ja lopuksi vapaamuotoiseen keskusteluun toiveista tulevaisuuden projektikommunikoinnista.

Henkilöt 1 ja 7 haastateltiin englannin kielellä käyttäen englanninkielistä kysymyslistaa, Liite C. Henkilöt 2 - 6 haastateltiin suomen kielellä käyttäen apuna suomenkielistä kysymyslistaa, Liite B.

Taulukko 1: *Haastateltavien taustojen kartoitus.*

	Koulutustausta ketterästä kehittämisestä?	Kokemus aiemmista ketteristä projekteista?
Henkilö 1	Scrum-tuoteomistajan koulutus, sertifioitu.	Toiminut projektipäällikkönä, Scrum-mestarina ja tuoteomistajana.
Henkilö 2	Scrum-mestarin koulutus.	Toiminut ketterien projektien taustalla ohjausryhmissä.
Henkilö 3	Yrityksen sisäinen koulutus ja itseopiskelua.	Toiminut tuoteomistajana muutamassa projektissa.
Henkilö 4	Itseopiskelua.	Toimii parhaillaan tuoteomistajana.
Henkilö 5	Kurssit yliopistolla ja itseopiskelua.	Toiminut useassa Scrum-projektissa kehittäjänä.
Henkilö 6	Tuoteomistajakoulutus, sertifioitu	Toimii parhaillaan tuoteomistajana.
Henkilö 7	Ei koulutusta.	Ei aiempaa kokemusta.

Näin haastateltavat kertoivat ketterän kehittämisen koulutustaustastaan, sekä kokemuksestaan ketteristä projekteista (taulukko 1). Osalla henkilöistä oli koulutustaustaa ketterään kehittämiseen, osalla ei. Osalta vastaajista löytyi myös käytännön kokemusta muista ketteristä projekteista. Yksi vastaajista osallistui ketterän ohjelmistoprojektin seuraamiseen nyt ensimmäistä kertaa.

Taulukko 2: *Tapauskysymykset ja vastaukset.*

	Miten olet osallistunut projektin seuraamiseen?	Oletko saanut tietoa sinulle tärkeistä asioista?	Oletko kyennyt saamaan kuvan projektin edistymisestä tähän mennessä?	Oletko kyennyt saamaan kuvan projektin tavoitteista?	Oletko kokenut kykeneväsi vaikuttamaan projektin lopputulokseen?
Henkilö 1	Demo, Intranet	Ei	Ei	Ei	Ei
Henkilö 2	Demo, Infosessio	Ei	Osittain	On	Osittain
Henkilö 3	Intranet, kollegat	Osittain	Osittain	On	Ei
Henkilö 4	Demo, Infosessio, Yammer	On	On	On	On
Henkilö 5	Demo, Jira, Slack	On	On	On	On
Henkilö 6	Demo, Infosessio, Jira, Slack	On	On	On	On
Henkilö 7	Infosessio, kollegat	On	On	On	On

Tapaustutkimuskysymysten vastauksissa oli selvää hajontaa. Henkilöt 1 ja 2 olivat selvästi tyytymättömiä projektin viestintään, eivätkä he kokeneet kykenevänsä vaikuttamaan projektiin riittävästi. Myös henkilö 3 oli osittain tyytymätön projektin viestintään. Henkilöt 4–7 olivat tyytyväisiä nykyiseen kommunikointitasoon ja pitivät sitä riittävänä.

Tyytymättömät henkilöt

Henkilöiden 1 ja 2 kohdalla yllättävä havainto oli, että heillä molemmilla on myös koulutustausta ketterän kehittämisen osalta. Henkilö 1 kertoi osallistuneensa satunnaisesti demo-tilaisuuksiin ja huomanneensa satunnaisesti uutisia yrityksen intra-verkossa. Henkilö 2 kertoi myös osallistuneensa satunnaisesti demo-tilaisuuksiin ja tämän lisäksi myös satunnaisesti infosessioihin.

Kysyttäessä projektin kulusta omin sanoin henkilö 1 ei osannut kertoa projektin iteraatioista, niiden sisällöstä tai projektin vaihteista mitään. Kysyttäessä ”Oletko kokenut kykeneväsi vaikuttamaan projektin lopputulokseen” henkilö 1 kertoi kuitenkin eräästä mieleen jääneestä tapauksesta:

”Lightning projektin yhteydessä kyselin, mitä CRM Lightning käyttäjille tulee tapahtumaan projektin yhteydessä? Kysymykseeni ei ikinä vastattu. En saanut vastausta. Kysyin tätä eräässä kokouksessa.”

Henkilö 1:llä ei ollut varsinaista vastuullista roolia projektissa.

Henkilö 2 kertoi omin sanoin projektin kulusta omin sanoin varsin pitkästi. Henkilö 2 kertoi projektin kulusta portfolion näkökulmasta, joka kuuluukin hänen rooliinsa yrityksen organisaatiossa. Henkilö 2 oli tyytymätön tapausprojektin ulosraportointiin:

”Näytti siltä, että burndown ei koskaan saatu nollaan, vaan aina jäi asioita seuraavaan sprinttiin. Raportoinnin puutteellisuus ja kommunikointi oli haastavaa. Portfolion kannalta ei nähty paljonko rahaa oli mennyt, paljonko oli jäljellä. Jossakin aiheessa tehtiin review Product Ownerin ja Business Ownerin kanssa 8 sprintin kohdalla ja todettiin että ¾ osaa sprinteistä on käytetty mutta MVP ei ole ¾ valmis. Ylemmän johdon kysymyksiin paljonko se maksaa ja milloin se on valmis ei osattu turvallisin mielin vastata.”

Osittain tyytyväiset henkilöt

Henkilö 3 toimi yhtenä tuoteomistajana projektin alkaessa, mutta oli siirtynyt uusiin työtehtäviin useampia kuukausia ennen haastattelua. Henkilö 3 kertoi olleensa pulassa projektin alussa vaatimusten eli storyjen kirjoittamisen kanssa ja olisi toivonut siihen enemmän tukea:

”Kenelläkään ei ollut Agile taustaa ja silti minun piti tehdä user stoorit ja epicit.”

Uusiin tehtäviin siirtymisen jälkeen henkilö 3 ei osallistunut demo-tapahtumiin tai infosessioihin.

Tyytyväiset henkilöt

Henkilöt 4-6 olivat osallistuneet aktiivisesti tapausprojektiin eri rooleissa. Aktiivisuus näkyi myös kysymysten vastauksissa, kaikki olivat tyytyväisiä projektin viestintään ja läpinäkyvyyteen.

Henkilö 4 kertoi olevansa tyytyväinen, että ketterän kehittämisen mallia otetaan käyttöön käyttöön kohdeyrityksessä. Henkilö 4 kertoi seuraavasti:

”Demoissa käydään läpi mitä on saatu aikaan. Demossa voisi vielä tehdä yhteenvedon mitä on tehty spare parts, mitä on tehty warranty.”

”Olen kokenut saavani tietää riittävästi minua kiinnostavista asioista. Toivoisin vielä skarpimmin aikatauluja ja päivämääriä, milloin minulta odotetaan jotakin.”

Henkilö 4 toimi projektissa asianomistajana. Henkilö 4 toivoi aktiivisempaa tietoitusta Yammer-ryhmässä. Kokonaisuutena henkilö 4 oli kuitenkin tyytyväinen kommunikointiin.

Henkilö 5 toimi projektitiimissä kehittäjänä ja arkkitehtina osallistuen aktiivisesti projektin demo-tilaisuuksiin. Henkilö 5 kertoi:

”Demoissa on tärkeää että stakeholderit ja businessin edustajat osallistuvat niihin ja antavat suoraa palautetta.”

Henkilö 5 koki saavansa riittävää kommunikaatiota asianomistajilta (stakeholder) iteraatiotyön keskellä:

”Teknisiä työkaluja on hyödynnetty hyvin, Slack, Jira, Skype.”

Henkilö 5 oli tyytyväinen projektin kommunikointiin.

Henkilö 6 toimi tuoteomistajana projektissa. Hänellä oli luonnollisesti erittäin aktiivinen rooli kaikessa projektin viestinnässä. Henkilö 6 kertoi osallistuvansa päivittäin dailyyn, sekä demoihin ja infosessioihin. Henkilö 6 piti projektin viestintää onnistuneena.

Henkilöllä 7 ei ollut vastuullista roolia projektissa. Hänellä ei ollut koulutustaustaa tai kokemusta ketterästä kehittämisestä. Kysyttäessä kertomusta omin sanoin projektin

kulusta hän ei osannut kertoa projektin eri vaiheista. Henkilö 7 oli kuitenkin osallistunut aktiivisesti infosessioihin, esitti niissä myös kysymyksiä ja antoi aktiivisesti palautetta. Henkilö 7 kertoi olevansa tyytyväinen loppuasiakkaalle tehtävästä palvelukehityksestä ja kertoi tämän olevan askel oikeaan suuntaan. Projektia henkilö 7 kertoi seuraavansa omasta mielenkiinnostaan.

Aktiivisuus eri viestintäkanavissa

Tutkimuksen yhteydessä kerättiin myös kvantitatiivista dataa eri viestintäkanavien käyttäjämääristä (taulukko 3). Tapaustutkimuksen kvalitatiiviset haastattelut suoritettiin 13.5.2019–3.6.2019 välisenä aikana. Tuona ajanjaksona järjestettiin iteraation 27 esittelykokous (demo), 24.5.2019. Kvantitatiivinen data kerättiin tämän iteraatioesittelyn yhteydessä.

Taulukko 3: *Aktiivisuus eri viestintäkanavissa iteraation jälkeen.*

Viestintäkanava	Seuraajamäärä
Iteraation esittelykokous (demo)	36
Iteraation jälkeinen infosessio	48
Iteraation tulosten julkaisu Yammerissa	118

6. ANALYYSI JA JOHTOPÄÄTÖKSET

Haastateltavat henkilöt edustivat yrityksen organisaatiota monella eri tasolla. Haastateltavilla oli myös erilaiset intressit projektia kohtaan.

Negatiivisesti suhtautuneiden henkilöiden voidaan katsoa omaavan kokemusta ketterän kehittämisen projekteista. Kumpikaan henkilöistä 1 ja 2 eivät osallistuneet aktiivisesti projektin demoihin tai infosessioihin. Nämä iteraation jälkeen pidettävät tapahtumat ovat olennainen osa ketterää kehittämistä, osoittamista mitä on saatu oikeasti aikaan.

Tapausprojektin viestintään tyytymättömät henkilöt

Henkilö 1:llä oli negatiivinen kokemus projektia kohtaan. Tämä on saattanut vaikuttaa yleiseen mielikuvaan itse projektista tai sen viestinnästä. Haastattelussa ei varsinaisesti saatu selville, miten projektin viestinnässä olisi voitu parantaa. Henkilö 1 toimi erään osa-alueen asianomistajana ja negatiivinen kokemus koski epäselvyyttä, milloin henkilölle 1 tärkeät tarinat tulevat tekemisen alle.

Tästä tilanteesta on tunnistettavissa haaste; ennen iteraation alkua Scrum-tiimi ei tiedä tulevan iteraation käyttäjätarinoita, ne päätetään vasta iteraation suunnittelukokouksessa. Scrum-prosessin demot on tarkoitettu valmiiden käyttötapauksen esittelyyn, niiden jotka on saatu valmiiksi. Asianomistajat ovat kuitenkin kiinnostuneita omien käyttötapauksensa etenemisestä. Tämä tuo painetta tuoteomistajalle, jonka vastuulla on käyttötapauksen priorisointi. Isossa organisaatiossa tuoteomistajalla on useita asianomistajia kuultavanaan, useita käyttötarinoita priorisoitavana (kuva 6).

Kyseistä tilannetta voi helpottaa, mikäli työjonot (Backlog) ovat hyvässä kunnossa ja priorisoitu. Toisaalta työjonon liian pitkälle työstimisessä on riski vaatimusten muuttumisesta ennen kuin käyttäjätarina pääsee itse iteraatioon. On tärkeää, että työjonon tärkeimmiksi priorisoidut käyttötapauksitarinat ovat valmiina seuraavan iteraation alkaessa. Matalammalle priorisoituja käyttäjätarinoita ei siis kannata tehdä valmiiksi asti. Scrumissa tuoteomistaja joutuu tasapainoilemaan työlistan kanssa.

Standish Groupin Chaos raportissa todettiin kokeneiden ja kyvykkäiden Scrum-tiimien onnistuvan projekteissaan huomattavasti todennäköisemmin kuin kokemattomat Scrum-tiimit (luku 2.4, Tutkimuksia ohjelmistotuotannon haasteista (kuva 5). Eräs onnistumisen elementti lienee tiimin auttaminen työlistan kanssa. Scrum-tiimi on itse vastuussa onnistumisesta. Kyvykäs ja kokenut tiimi tukee tuoteomistajaa työlistan pitämisessä selkeänä, priorisoituna ja työmäärät oikein arvioituna.

Kommunikointi tuoteomistajan ja asianomistajien välillä on haasteellista suuressa organisaatiossa, jossa henkilöiden toimipaikat ovat maantieteellisesti hajautettuja. Vaikka jokainen olisi tavoitettavissa toimistotyötuntien puitteissa videopuhelulla, kynnys yhteyden ottamiseen kasvaa.

Läpinäkyvyys on eräs tärkeä Scrumin arvo. Scrumissa fokus on iteraatioissa ja priorisoiduissa työjonoissa. Vaikka työjonot olisivatkin nähtävissä, ilmeisen harva niitä kuitenkaan aktiivisesti tutkii. Täten tulisi harkita, että iteraation sen hetkinen sisältö viestittää erikseen iteraation aikana. Tällä voisi olla useita positiivisia vaikutuksia:

- Luo paineen pitää iteraation työjono (Backlog) hyvässä kunnossa.
- Iteraatioon otetaan vain sen verran käyttäjätarinoita kuin kyetään saamaan valmiiksi.
- Luo paineen arvioida käyttäjätarinan työmääräarvion mahdollisimman tarkaksi.
- Luo odotusarvon iteraation demoön valmiista käyttäjätarinoista.
- Luo paineen saada käyttäjätarinoita valmiiksi iteraation demoön.
- Parantaa läpinäkyvyyttä ja luottamusta Scrum-tiimiin.

Henkilöllä 1 oli pääsy ja mahdollisuus näkyvyyteen eli iteraatioiden työjonoihin ja niitä sisältäviin käyttötapaustarinoihin. Tätä mahdollisuutta hän ei kuitenkaan ollut käyttänyt.

Henkilö 2 kertoi tarkastelevansa projektia portfolion näkökulmasta. Henkilö 2 kertoi myös olevansa tyytymätön portfolion näkökulmaan; on tärkeää, että rahaa sijoittanut tietää mitä hän sijoituksellaan saa ja milloin. Haastateltava totesi, että:

”Koko portfolion kannalta haaste alkuvaiheessa oli tiedon saaminen, edistyykö projekti, sen tiedon saaminen verrattuna siihen mikä oli tavoite, jouduttiin perehtymään agiiliin raportointiin. Tämä oli yksi asia mikä projektissa olisi pitänyt paremmin testata agiilit työkalut, Jira ja Confluence olisivat mahdollistaneet raportoinnin paremmalla tasolla.

Mutta tuntiraportteja ei tehty ja burndown chartteja ei saatu. Näytti siltä, että burndown ei koskaan saatu nollaan vaan aina jäi asioita seuraavaan sprinttiin. Raportoinnin puutteellisuus ja kommunikointi oli haastavaa. Portfolion kannalta ei nähty paljonko rahaa oli mennyt, paljonko oli jäljellä.”

Tapausprojektin alkuvaiheessa projektitiimissä tehtiin päätös olla kirjaamatta tunteja itse käyttötapausten tasolle. Itse ohjelmointityö teetettiin ulkopuolisilla yrityksillä. Tuoteomistaja keräsi työtunnit kootusti laskuilta. Projektille oli myönnetty budjetti useammaksi kuukaudeksi eteenpäin. Henkilö 2 oli siis tietoinen portfolion budjetista, mutta kuitenkin ei tarkasti portfolion kehittämisen sisällöstä.

Pitääkö iteraation lopulla burndown charin olla nolla, on hyvä kysymys. Iteraatioon ei pitäisi valita enempää tai isompia käyttötapauksia kuin mitä kyetään saamaan tehdyksi. Toisaalta käytännössä isommassa, useamman henkilön tiimissä työ usein jakautuu eri vastuualueisiin. Käyttötapauksia on usein hankalaa pilkkoa usealle tekijälle tehtäväksi. Tekijöiden määrän lisääminen ei lisää valmista lopputulosta samassa suhteessa. Tätä varten on tärkeää, että työlista on hyvässä kunnossa ja priorisoitu. Mikäli iteraatiossa osa tiimin henkilöistä on saanut työnsä valmiiksi, nostetaan iteraatioon uusi käyttötapaus, mielellään kooltaan sellainen, joka voisi valmistua jo itse iteraation aikana. Tämä voi kuitenkin johtaa tilanteeseen, jossa iteraatioon valitut käyttötapaukset eivät valmistukaan suunnittelussa aikataulussa.

Toinen mahdollinen tekijä, miksi burndown chart ei ole nolla iteraation päätteeksi, on tiimin kokemattomuus (kuva 2). Haastatteluja tehdessä projektin alusta oli kulunut noin vuosi. Tähän vuoteen oli mahtunut paljon henkilövaihdoksia. Kokematon tiimi tekee helposti useanlaisia virheitä;

- Käyttötapausten työmäärä arvioidaan väärin,
- Käyttötapausten monimutkaisuus arvioidaan väärin,
- Käyttötapaukset eivät ole riittävän selkeästi ja yksiselitteisesti kirjoitettu,
- Käyttötapausta ei osata jakaa pienempiin osiin, jotta se voitaisiin suorittaa yhdessä iteraatiossa,
- Tiimistä puuttuu teknologinen osaaminen käyttötapausten tekemiseen, ja
- Scrum-mestari ei osaa ohjata tiimiä oikeaan suuntaan.

Henkilöiden vaihtuminen ketterässä tiimissä on aina ongelmallista. Tiimin dynamiikka kärsii ja henkilöiden luottamus toisiinsa pitää rakentaa uudelleen.

Henkilö 2 kertoi haastattelussa myös näin;

”Jossakin aiheessa tehtiin review Product Ownerin ja Business Ownerin kanssa 8 sprintin kohdalla ja todettiin että $\frac{3}{4}$ osaa sprinteistä on käytetty mutta MVP ei ole $\frac{3}{4}$ valmis.”

Agilen ja Scrumin idea ei ole kiveen hakattu projektin määrittely. Mikäli projektin päämäärä, suunnitelma, on jo ennalta päätetty, että tämä projekti on syytä viedä läpi vesiputousmallia noudattaen, projektissa ei ole tilaa ketterän kehittämisen luovuudelle. Ketterässä kehittämisessä resursseja sijoitetaan tuotteen kehittämiseen, ei projektiin, jolla on alku ja loppu.

Tuoteomistajalla pitää olla selkeä visio, johon portfolion näkökulmasta sijoitetaan. Mikään ohjelmisto ei ikinä tule olemaan täysin valmis, vaan portfolioon, ja sen sisältämiin ohjelmistoihin joudutaan usein investoimaan koko niiden elinkaaren ajan. Portfolion näkökulmasta investoinnin tekijän pitää kuitenkin voida luottaa, että vision kannalta kriittiset ominaisuudet saadaan varatulla resurssilla tehdyksi.

Henkilö 2:n mietteet ja ajatukset portfolion näkökulmaan ovat hyviä huomioita. Scrumissa keskitytään usein vain viimeisimmän iteraation aikaansaannoksiin. Mikäli nämä aikaansaannokset linkitettäisiin selkeästi kehitysaihio-tasolle, ja kehitysaihio olisi hyvin ennalta mietitty ja sisällöltään stabiili, kyettäisiin luotettavammin antamaan myös iso kuva projektin etenemisestä.

Yhteistä tyytymättömille henkilöille oli; he eivät osallistuneet aktiivisesti projektin viestintäkokouksiin. Perusteellista vastausta sille, miten suuressa organisaatiossa jokainen henkilö saataisiin osalliseksi projektille ominaisesta tyylistä projektiviestinnässä, ei saatu. Suurissa globaaleissa organisaatioissa on aina käynnissä useita projekteja samanaikaisesti. Samasta huomiosta kamppailee siis moni projekti ja viestintäkanava. Tärkeintä lienee, että enemmistö projektia aktiivisesti seuraavista henkilöistä on tyytyväisiä. Ja mikäli se on pienellä vaivannäöllä mahdollista, niin viestintää voi pyrkiä kehittämään yksittäisiä näkökulmia parantavaan suuntaan.

Tutkimuksen yhteydessä kerättiin myös tietoa käyttäjämääristä eri viestintäkanavissa (taulukko 3). Tulos osoittaa, että ylivoimaisesti suosituin tapa seurata projektia

kohdeyrityksessä on Yammer. Yammer on viestintäkanavana kuitenkin hyvin erilainen kuin kaksi muuta viestintäkanavaa ja tapaa.

Iteraation esittelykokous ja infosessio ovat kalenteriin varattuja tapahtumia. Henkilöt 4 ja 7 piti kalenterivarausta hyvänä asiana. Näin projektin seuraamiselle on varattu aika muun työkiireen keskellä. Henkilö 7 piti myös vuorovaikutusmahdollisuudesta infosessioissa. Vuorovaikutusmahdollisuutta ja varattua aika kalenterista voidaan siis pitää infosession ja iteraation esittelykokouksen etuina.

Yammerin etuna voidaan taas pitää vapautta aikavyöhykkeistä ja kalenteriajoista. Globaali organisaatio sisältää aina ihmisiä usealla eri aika vyöhykkeellä. Yhteisen, kaikille sopivan kalenteriajan löytyminen on usein siis lähes mahdottomuus. Tällaisessa tilanteessa Yammer toimii oivana viestinviejänä. Henkilö 4 piti mielekkäänä seurata projektia myös Yammerin välityksellä.

Tutkimusta varten ei kyetty saamaan tietoa, osallistuvatko Yammeria aktiivisesti käyttäneet myös infosessioihin ja/tai esittelykokouksiin. Oletettavaa kuitenkin on, että Yammerin kautta projektia aktiivisesti seuraavista valtaosa seuraa projektia passiivisesti, olematta varsinaisesti vuorovaikutuksessa projektin kanssa.

7. YHTEENVETO

Tämän tutkimuksen tavoite oli selvittää tapaustutkimuksen aiheomistajien, ja muiden kohdeorganisaatioissa projektista kiinnostuneiden, tyytyväisyyttä tapaustutkimuksen projektiviestintää kohtaan. Projektia suoritettiin Scrumia hyödyntäen ja tapausprojekti oli eräs ensimmäisistä ketterän kehittämisen projekteista kohdeyrityksessä. Kohdeyritys oli kansainvälisesti tunnettu teknologiateollisuuden yritys. Koska yrityksen organisaatio toimii globaalisti, asettaa se haasteita kommunikoinnille. Eräs ketterän kehittämisen tärkeimmistä arvoista on avoimuus ja läpinäkyvyys. Tämän vuoksi tutkimus avoimuudesta ja viestinnän onnistumisesta oli kiinnostava. Tutkimuksesta myös toivottiin käyvän ilmi, mitkä viestintätavat ovat toimivia ja mitkä eivät.

Tutkimus suoritettiin kahdenkeskisten haastattelujen avulla. Haastattelut aloitettiin, kun projekti oli ollut käynnissä noin vuoden ajan. Iteraatioita oli sillä hetkellä suoritettu yhteensä 26 kertaa. Ajankohta haastatteluille katsottiin sopivaksi, sillä projektin rutiinit olivat alkaneet vakiintumaan. Haastateltaviksi valittiin sopiviksi katsottuja henkilöitä, joilla tiedettiin olevan jokin yhteys ja mielenkiinto tapausprojektia kohtaan. Haastatteluihin valittiin henkilöitä eri puolilta ja tasoilta organisaatiota. Haastattelut käytiin läpi puolistrukturoidusti valmista kysymyslistaa hyödyntäen, keskustelua kuitenkin rajoittamatta.

Tapausprojektin tuloksista voitiin päätellä; mitä aktiivisempi rooli henkilöllä oli projektiin nähden, sitä tyytyväisempiä he olivat myös kommunikointiin ja avoimuuteen. Haastatelluista 4/7 oli tyytyväisiä tapausprojektin kommunikointiin. Nämä henkilöt, jotka osallistuivat aktiivisesti projektin iteraatioiden demoihin ja niitä seuraaviin infosessioihin, pitivät tiedonsaantia riittävänä ja onnistuneena. Nämä haastatellut henkilöt kokivat voivansa vaikuttaa projektin lopputulokseen.

Henkilöillä 1 ja 2 löytyi ketterän kehittämisen koulutustaustaa. Henkilöt olivat taustoiltaan muutenkin poikkeavia muihin haastateltaviin nähden; heillä ei ollut suoraa ja aktiivista roolia tapausprojektissa. Henkilön 2 vastuualue oli ylempänä organisaation tietohallinnossa, eikä hän osallistunut aktiivisesti projektin viestintäkanaviin. Henkilö 2 ei kokenut saavansa itselleen tärkeitä tietoja projektin viestinnästä: *”Ylemmän johdon kysymyksiin paljonko se maksaa ja milloin se on valmis ei osattu turvallis in mielin*

vastata.”. Projektin iteraatitasolla tähän kysymykseen ei yritettykään vastata. Projektin viestinnässä kerrottiin lähinnä sen hetkistä iteraation saavutuksista. Tässä kohdassa lienee ollut tietokatos projektissa ja sen viestinnässä. Scrumissa itsessään keskitytään hallittujen iteraatioiden pyörittämiseen. Työlistan priorisointi, vision ja mission eteenpäin vieminen sekä niistä kommunikointi, kuuluu tuoteomistajan vastuulle. Tilanteen haasteellisuutta voi hahmottaa kuvan 9 avulla, ketterän yrityksen konsepti (sivu 26). Scrum ja iteraatiot toimivat projektitasolla ja iteraatioiden esittelyissä esitellään kulloisenkin iteraation valmiita komponentteja ja tuotoksia. Nämä tuotokset ovat pieniä palasia portfolion visiosta. Kuitenkin seuraamalla yksittäisiä iteraatioiden esittelyjä, voi olla vaikeaa saada kokonaiskuvaa projektin etenemisestä. Tuoteomistajien tulisikin käydä aktiivista keskustelua portfoliosta ja visiosta muun organisaation kanssa. Mikään ei toki myöskään estä esittelemästä kokonaiskuvaa iteraation esittelyn yhteydessä. Tämä voitaisiin tehdä esimerkiksi joka 5. iteraation esittelyn yhteydessä.

Henkilö 2 on kiinnostunut tiedosta, milloin kehitysaihiot (epic) ovat valmiit ja tietty MVP versio on valmis. Projektissa ja sen viestinnässä keskityttiin kuitenkin yksittäisten tarinoiden valmistumiseen. Kehitysaihion katsottiin olevan valmis, kun tuoteomistaja on sen hyväksynyt. Tapausprojektin product owner, henkilö 7: *”Stakeholderit ovat ottaneet suoraan yhteyttä, kun tekemisessä on riippuvuuksia, tai kun on ilmennyt monimutkaisuuksia ja kehittäminen ei etene suunnitellussa aikataulussa.”*. Jos kehitysaihion sisältö on joustava, myös aikataulu venyy. Tämä pitää aiheomistajat tyytyväisenä, kuten haastattelun tuloksista voidaankin päätellä. Projektia isommassa kuvassa seuraavalle, esimerkiksi portfoliosta vastaavalle, tämä voi kuitenkin olla ongelmallista, koska kokonaisuus (milestone) ei valmistukaan suunnitellussa aikataulussa. Jos kehitysaihion sisältö on joustava, on sen työmäärä myös joustava ja työmäärän arviointi todennäköisesti epäonnistuu.

”Tuoteomistajan ei esimerkiksi pidä lähtökohtaisesti lisätä kehitysjonoon loppukäyttäjien jokaista toivetta. Jos lista paisuu, tuleeko lopputuotteesta silloin strategian mukainen ratkaisu vai kokoelma kivoja ominaisuuksia?”. Näin kiteyttää Turun yliopiston tutkija Altti Lagstedt osuvasti eräässä lehtihaastattelussa [26].

Kehitysaihioiden kokoa tulisi siis pilkkoa pienemmiksi, jotta työmäärän ja edistymisen seuraaminen olisi ollut helpompaa. Projektihenkilöiden kokemus ja kyvykkyys olisi kenties myös ohjannut kehitysaihion sisällön rajoittamiseen ja stabiloimiseen. Tämä on

yksi lisätutkimismahdollisuus; käyttötapaustarinoiden koon vaikutus niiden valmistumiseen suunnitellussa aikataulussa.

Henkilön 1 kohdalla tyytymättömyyden syystä ei saatu täyttä varmuutta. Ehkä taustalla oli muutamat negatiiviset kokemukset tapausprojektissa, joka sai koko projektille negatiivisen leiman. Henkilö 1 ei osallistunut aktiivisesti projektin seuraamiseen.

Chaos raportin mukaan projektitiimin kyvykkyydellä ja kokemuksella on selkeä vaikutus ohjelmistoprojektin onnistumiseen (sivu 15). Tapaustudkimuksen projektia ei voida pitää erityisen kokeneena ketterän kehittämisen, tai Scrumin suhteen. Yrityksen organisaatiossa ketterä kehittäminen oli käytännön tasolla ylipäättänsä sangen uutta. Kokemuksen myötä projektissa työskennelleet henkilöt, ja yrityksen organisaatio yleisesti, saa kuitenkin arvokasta kokemusta. Tutkimuksen tulosten avulla organisaatiolla on nyt mahdollisuus oppia ja kehittää kyvykkyyttään. Tämän tutkimuksen perusteella kehitysaihiot (epic) tulisi sitoa paremmin yhteen portfolion vision kanssa.

Tämä tapaustudkimus perustuu yhteen hetkeen, jolloin haastattelut suoritettiin. Haastatteluihin vaikuttavat mielikuvat itse projektista, sekä henkilökohtaiset kokemukset aiemmista projekteista. Tutkimuksen tulosta ei voida täten pitää yleistettävänä. Tutkimuksessa selvisi, että viestintään tyytyväisimmät henkilöt osallistuivat aktiivisesti tarjottuihin viestintäkanaviin. Tämä tuntuu myös loogiselta. Tutkimus tarjoaa kuitenkin ajatuksia, miten kehittää ketterän projektin viestintää ja avoimuutta suuressa organisaatiossa.

Tutkimuksessa selvisi myös, että globaalissa organisaatiossa tulee tarjota useita eri viestintäkanavavaihtoehtoja, sillä jokaisella kanavalla ja tavalla on omat kannattajansa. Esimerkiksi Yammer osoittautui käyttäjämäärissä mitattuna erääksi tärkeimmistä viestintäkanavista. Scrum ei itsessään näitä viestintäkanavia tarjoa, vaan ne on jokaisen organisaation itse kehitettävä ja tarpeeseen muokattava.

Onkin hyvä päättää Scrum-kouluttaja ja -valmentaja Lare Lekmanin toteamukseen; *”Scrum ei itsessään ole täsmällinen prosessi. Se antaa väljähkön, mutta tarkkaan harkitun viitekehyksen, jonka puitteissa toimia”* [26].

LÄHTEET

- [1] Manifesto for Agile Software Development, viitattu 12.1.2019. Saatavissa: <https://agilemanifesto.org/iso/fi/principles.html>
- [2] Ilkka Haikala, Jukka Märijärvi: Ohjelmistotuotanto, Talentum, 2004.
- [3] Standish Group International Inc, Chaos report 2015. Saatavilla: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- [4] Leffingwell, D., Widrig, D., Managing software Requirements: A use case approach, Addison-Wesley, 2003.
- [5] Van Vliet, H., Software engineering: Principles and practice, Wiley Publishing, 2008.
- [6] Jim Highsmith, History: The Agile Manifesto, 2001. Saatavilla: <https://agilemanifesto.org/history.html>
- [7] Petri Kettunen, Maarit Laanti, Combining agile software projects and large-scale organizational agility, Software process improvements and practice, John Wiley & Sons Ltd, 2007.
- [8] Hiroshi Katayama, David Bennett, Agility, adaptability and leanness: a comparison of concepts and a study of practice, International journal of production economics, Vol.60/61, No.3, 1999.
- [9] Peter Gould, What is agility? IEE Manufacturing Engineering, Vol.76, No.1, 1997.
- [10] I. Sommerville, software engineering, Pearson Education Limited, 2004.
- [11] Jeff Sutherland, Anton Viktorov, Jack Blount, Nikolai Puntikov, Distributed Scrum: Agile project management with outsourced teams, HICSS'07 40th Hawaii International Conference, 2007, s.1-10.
- [12] Julian M. Bass, Scrum master activities: Process tailoring in large enterprise projects, OpenAIR@GRU, 2014, s.1-12.
- [13] Dean Leffingwell, Scaling software agility – the big picture of enterprise agility, whitepaper, 2009. Saatavilla: <https://scalingsoftwareagility.files.wordpress.com/2007/03/the-big-picture-of-enterprise-agilitywhitepaper.pdf>
- [14] Jurgen Appelo, Management 3.0 Leading agile developers, developing agile leader, Addison-Wesley, 2011.
- [15] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, Juhani Warsta, Agile software development methods: Review and analysis, VTT publications 478, 2002. Saatavilla: <https://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>

- [16] Ken Schwaber, Mike Beedle, Agile software development with Scrum, Prentice hall, 2002.
- [17] Michael Hron, Nikolaus Obwegeser, Scrum in practice: an overview of Scrum adaptations, Proceedings of the 51st Hawaii International Conference on System Sciences, 2018. Saatavilla: <http://hdl.handle.net/10125/50568>
- [18] Michael James, About Scrum, Viitattu 26.1.2019. Saatavilla: <http://scrumreferencecard.com/>
- [19] David Parnas, Agile methods and GSD: The wrong solution to an old but real problem, Communication of the ACM, 49(10), 2006, s.28-29.
- [20] Akif, R., H. Majeed, Issues and Challenges in Scrum implementation, International Journal of Scientific & Engineering research, Volume 3 Issue 8, 2012, s.1-3.
- [21] Graig Larman, Bas Vodde, Large-Scale Scrum, Addisson-Wesley, 2017.
- [22] Van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., and Bijlsma, L., Towards a Reference Framework for Software Product Management, RE'06:14th IEEE International Requirements Engineering Conference, 2006.
- [23] Maarit Laanti, Agile methods in large-scale software development organizations, 2012.
- [24] Jutta Eckstein, Agile software development in the large: diving into the deep, Dorset house publishing, 2004.
- [25] Robert K. Yin, Case study research, design and methods, Sage Publications, Thousand Oaks, 1994.
- [26] Kari Ahokas, Scrum – miksi se on niin vaikea hallita? Suomen suosituin ketterä menetelmä on liian usein taikasana, jonka alle ”piilotetaan omat paskat prosessit”, Mikrobitti, Alma talent 4.1.2020. Saatavilla: <https://www.mikrobitti.fi/uutiset/scrum-miksi-se-on-niin-vaikea-hallita-suomen-suosituin-kettera-menetelma-on-liian-usein-taikasana-jonka-alle-piilotetaan-omat-paskat-prosessit/566c7d32-f271-4267-98ea-b407342eb774>

LIITE A: 12 ALKUPERÄISTÄ TEESIÄ AGILE MANIFESTO ARVOJEN TAKANA

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

LIITE B: HAASTATTELUN MUISTILISTA SUOMEKSI

Taustan kartoitus

- Mikä on asemasi organisaatiossa?
- Koulutustaustasi ketterästä kehittämisestä.
- Aiempi kokemuksesi ketteristä projekteista?
- Miksi tämän projektin seuraaminen on sinulle tärkeää?

Kerro omin sanoin projektin kulku alusta MVP vaiheeseen ja ensijulkaisuun asti.

Tapaustutkimus

- Miten olet osallistunut projektin seuraamiseen?
- Oletko saanut tietoa sinulle tärkeistä asioista?
- Oletko kyennyt saamaan kuvan projektin tähänastisista saavutuksista?
- Oletko kyennyt saamaan kuvan projektin tavoitteista?
- Oletko kokenut kykeneväsi vaikuttamaan projektin lopputulokseen?

Tulevaisuus

- Mikä olisi mieluisin tapa sinulle seurata sinulle tärkeitä asioita?
- Muita tutkimusaiheeseen liittyviä kommentteja tai huomioita.

LIITE C: HAASTATTELUN MUISTILISTA ENGLANNIKSI

Prestudy

- What is your position in the organization?
- What is your training history in agile development?
- What is your experience history in agile projects?
- Why is it important to you to follow this case-project?

In your own words, describe the case-project from the beginning till the MVP was published.

Case-study

- How have you participated and followed the case project?
- Have you been able to get info from the things that are important to you?
- Have you been able to get info from the project progress so far?
- Have you been able to get idea of the mission of the project?
- Have you been able to affect to the scope of the project?

Future

- What would be the most suitable way to you to follow the things that are important to you?
- Other comments regarding the project communication.